Data Cleaning, Outliers, Feature Engineering

Data Cleaning

You cannot fit machine learning models on raw data directly, because:

- Implementations require data to be numeric
- Algorithms impose specific requirements
- Raw data contains errors
- Columns may be redundant or irrelevant
- These problems require specialized techniques including:

...**Data Cleaning** to delete duplicate rows are redundant columns
...**Outlier Detection** and removal
...**Missing Value** identification and imputation
...**Feature Selection** with statistics and models
...**Feature Importance** with models
...**Data Transforms** to change data scales, types, and distributions
...**Dimensionality Reduction** to create low-dimensional projections

Common Data Quality Issues

Usually following use cases are experienced when preparing data for your machine learning model:

- There might be missing or erroneous values in the data set
- There might be categorical (Textual, Boolean) values in the data set and not all algorithms work well with textual values.
- Some features might have larger values than others and are required to be transformed for equal importance.
- Sometimes data contains a large number of dimensions and the number of dimensions are required to be reduced.

Missing Value Treatment

Why missing values treatment is required?

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behaviour and relationship with other variables correctly. It can lead to wrong prediction or classification.

Detecting and handling missing values in the correct way is important, as they can impact the results of the analysis, and there are algorithms that can't handle.

In **list wise deletion**, we delete observations where any of the variable is missing. Simplicity is one of the major advantage of this method, but this method reduces the power of model because it reduces the sample size.

Reasons of Missing Data

Missing Completely at Random (MCAR)

Missing at Random (MAR)

Missing Not at Random (MNAR)

Why Data Goes Missing?

Lets understand some of the reasons why data goes missing?

- **Missing At Random (MAR)** : Propensity for a data point to be missing is not related to missing data but its related to some of the observed data.

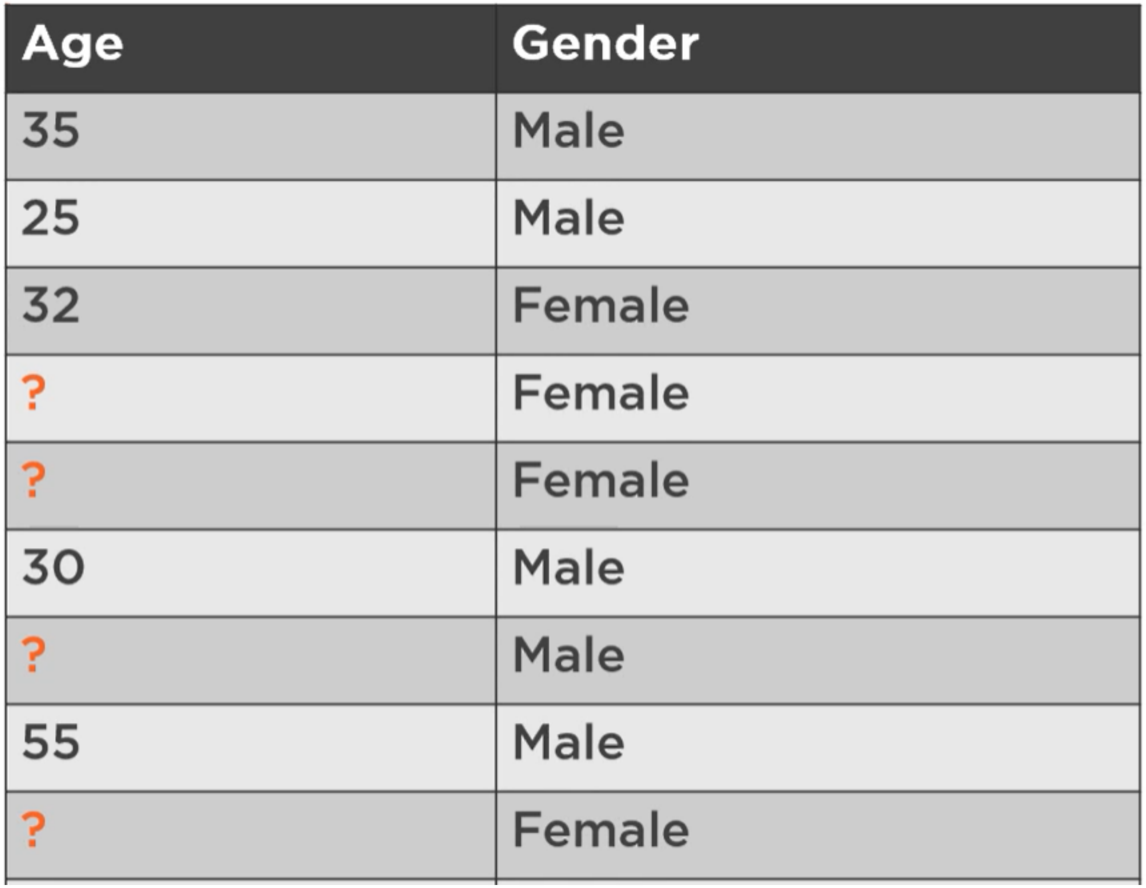
- **Missing Completely at Random (MCAR)** : The fact that the missing value is not related to observed data nor to its hypothetical value. In this case, the missing ness on the variable is completely unsystematic. There's no relationship between whether a data point is missing and any values in the data set, missing or observed.

- **Missing Not at Random (MNAR)** : Missing value either depends on the hypothetical value or some other variable's value. Example: Some Very High or Very Low people don't like to share their Salary; Ladies don't want to reveal their ages.

Missing Completely at Random (MCAR)

Country	Degree
United States	Bachelors
Cambodia	Masters
India	Preschool
Mexico	Bachelors
?	Masters
Germany	Doctorate
?	Masters
England	9th
Italy	11th

Missing at Random (MAR)



Age	Gender
35	Male
25	Male
32	Female
?	Female
?	Female
30	Male
?	Male
55	Male
?	Female

Missing Not at Random (MNAR)

Occupation
Priv-house-serv
Handlers-cleaners
Armed-Forces
?
?
Farming-fishing
?
Other-service
Exec-managerial

Missing Completely at Random (MCAR)

Definition: The probability of an instance being missing does not depend on known values or the missing value itself.

***Example*:** A data table was printed with no missing values and someone accidentally dropped some ink on it so that some cells are no longer readable. Here, we could assume that the missing values follow the same distribution as the known values.

Missing at Random (MAR)

Definition: The probability of an instance being missing may depend on known values but not on the missing value itself.

Sensor Example: In the case of a temperature sensor, the fact that a value is missing doesn’t depend on the temperature, but might be dependent on some other factor, for example on the battery charge of the thermometer.

***Survey example**:* Whether or not someone answers a question - e.g. about age- in a survey **doesn’t** depend on the answer itself, but may depend on the answer to another question, i.e. gender female.

Not Missing at Random (NMAR)

Definition**:** the probability of an instance being missing could depend on the value of the variable itself.

Sensor example: In the case of a temperature sensor, the sensor doesn’t work properly when it is colder than 5°C.

Survey example: Whether or not someone answers a question - e.g. number of sick days - in a survey **does** depend on the answer itself .

Only the knowledge of the data collection process and the business experience can tell whether the missing values we have found are of type MAR, MCAR, or NMAR.

We focus only on MAR or MCAR type of missing values. Imputing NMAR missing values is more complicated, since additional factors to just statistical distributions and statistical parameters have to be taken into account.

Detecting Missing Values and their Type

Before trying to understand where the missing values come from and why, we need to detect them. Common encodings for missing values are n/a, NA, -99, -999, ?, the empty string, or any other placeholder. When you open a new dataset, without instructions, you need to recognize if any such placeholders have been used to represent missing values.

Histograms are a great tool to find the placeholder character, if any.

For numerical values many datasets use a value far away from the distribution of the data to represent the missing values. A classic is the -999 for data in the positive range.

Usually, for nominal data, it is easier to recognize the placeholder for missing values, since the string format allows us to write some reference to a missing value, like “unknown” or “N/A”. The histogram can also help us here. For nominal data, bins with non fitting values could be an indicator of the missing value placeholder.

Dealing with Missing Values

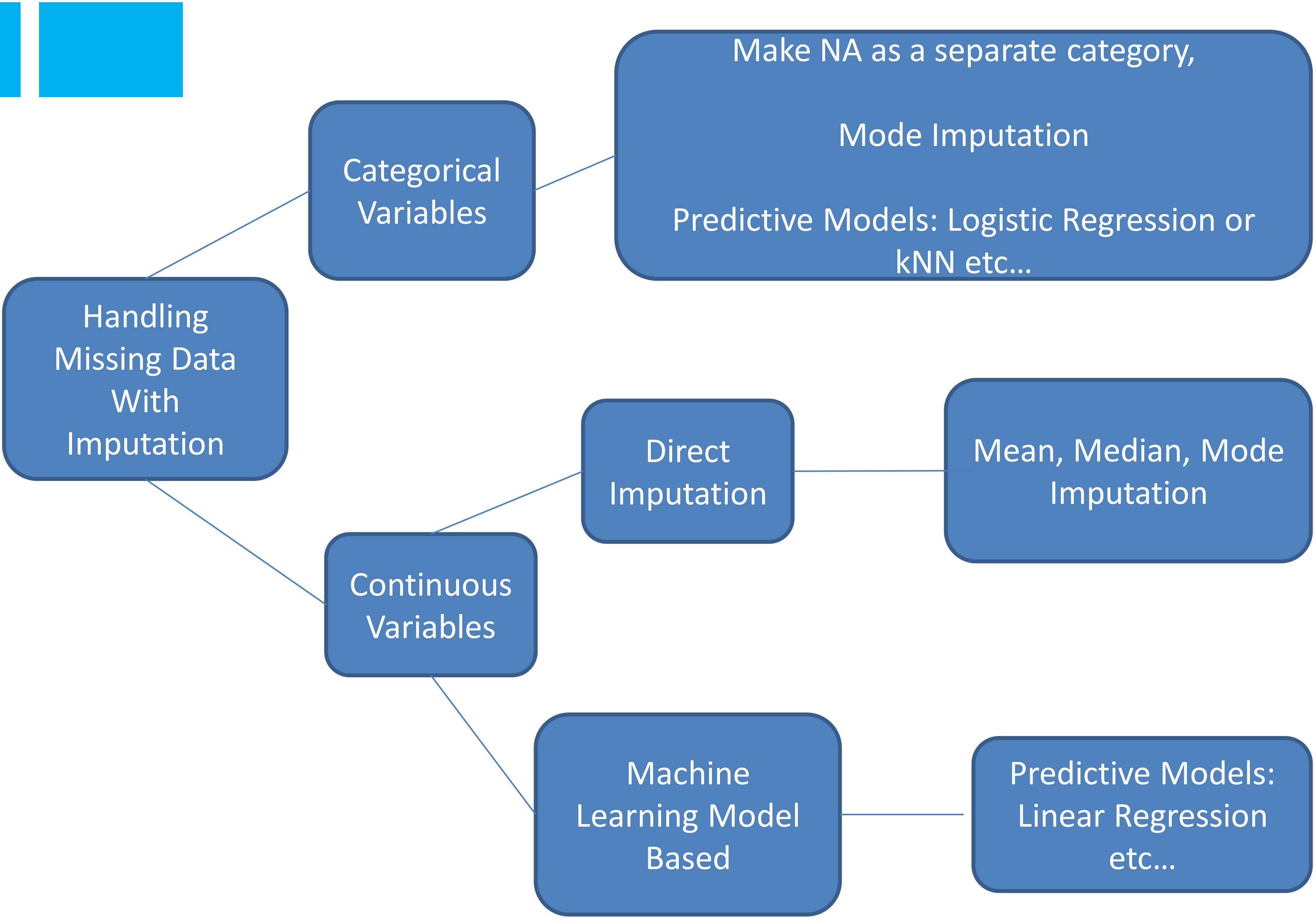
How to deal with missing values:

Set values as missing values: Identify values that indicate missing data, and yet are not recognized by the software as such, e.g treat blank strings, "NA", "XX", "999", etc. as missing.

You should try to get information from reliable external sources as much as possible, but if you can’t, then it is better to keep missing values as such rather than exaggerating the existing rows/columns.

Delete rows, columns: Rows could be deleted if the number of missing values are insignificant in number, as this would not impact the analysis. Columns could be removed if the missing values are quite significant in number.

Fill partial missing values using business judgement: Missing time zone, century, etc. These values are easily identifiable.



Handling Missing Data With Imputation

Categorical Variables

Make NA as a separate category,

Mode Imputation

Predictive Models: Logistic Regression or kNN etc…

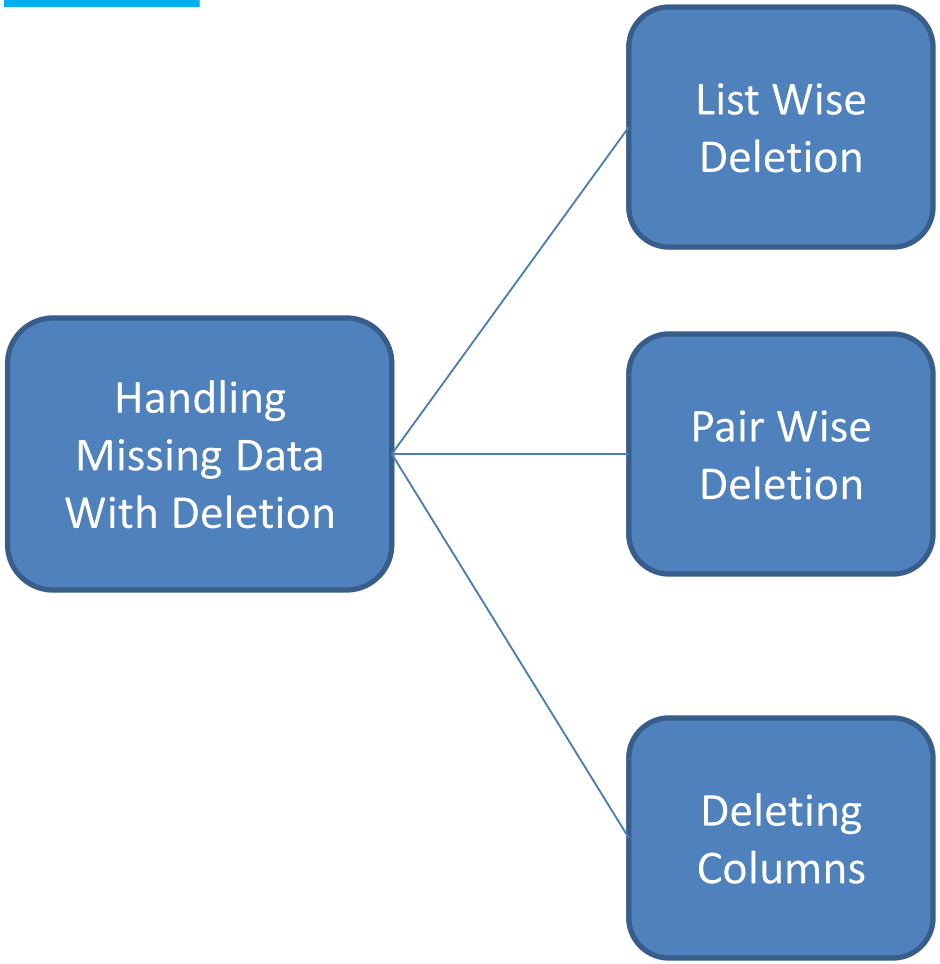
Continuous Variables

Direct Imputation

Mean, Median, Mode Imputation

Machine Learning Model Based

Predictive Models: Linear Regression etc…



Handling Missing Data With Deletion

List Wise Deletion

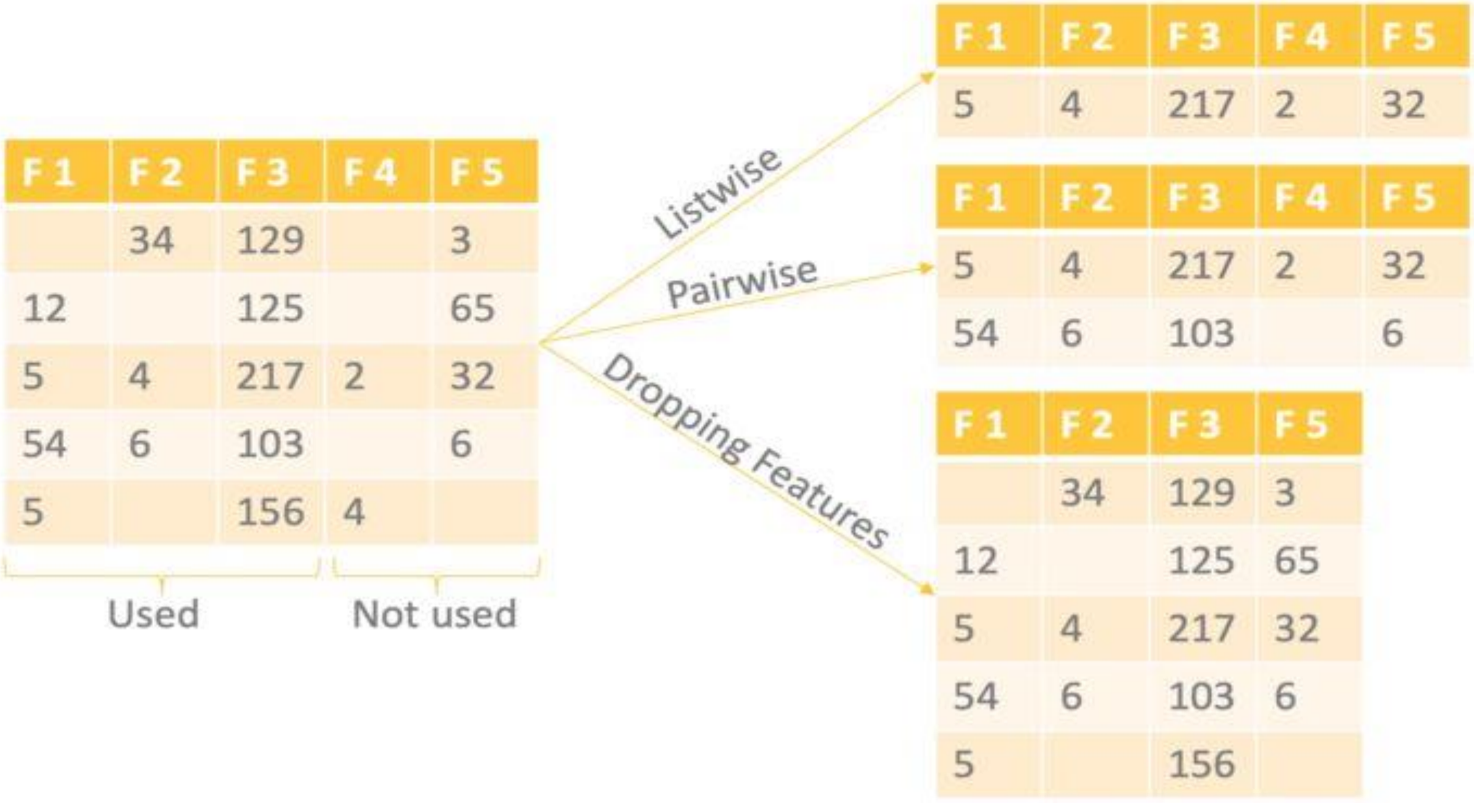
Pair Wise Deletion

Deleting Columns

Listwise Deletion: Delete all rows where one or more values are missing.

Pairwise Deletion: Delete only the rows that have missing values in the columns used for the analysis. It is only recommended to use this method if the missing data are MCAR.

Dropping Features: Drop entire columns with more missing values than a given threshold, e.g. 60%



F1	F2	F3	F4	F5
	34	129		3
12		125		65
5	4	217	2	32
54	6	103		6
5		156	4	

Listwise

F1	F2	F3	F4	F5
5	4	217	2	32

Pairwise

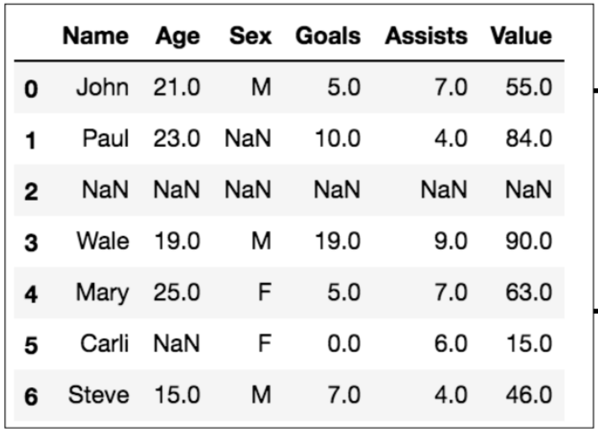
F1	F2	F3	F4	F5
5	4	217	2	32
54	6	103		6

Dropping Features

F1	F2	F3	F5
	34	129	3
12		125	65
5	4	217	32
54	6	103	6
5		156	

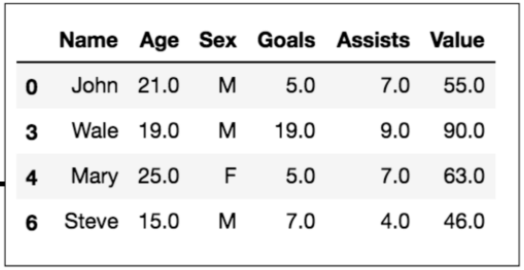
Researchers using list wise deletion will remove a case completely if it is missing a value for one of the variables included in the analysis. For example, say you are conducting analyses using cumulative high school GPA, hours of study for first semester, SAT score, and first semester grade in college algebra.
Participant X is missing data for cumulative high school GPA, therefore, Participant X will be completely removed from the analyses because the participant does not have complete data for all the variables.

Researchers using **pairwise deletion** will not omit a case completely from the analyses. Pairwise deletion omits cases based on the variables included in the analysis. As a result, analyses may be completed on subsets of the data depending on where values are missing. For the example listed above, **Participant X will be omitted from any analyses using cumulative high school GPA, but they will not be omitted from analyses for which the participant has complete data.**



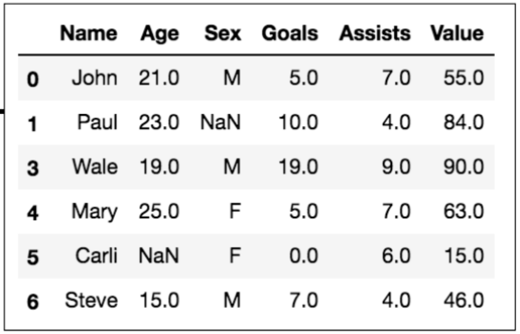
	Name	Age	Sex	Goals	Assists	Value
0	John	21.0	M	5.0	7.0	55.0
1	Paul	23.0	NaN	10.0	4.0	84.0
2	NaN	NaN	NaN	NaN	NaN	NaN
3	Wale	19.0	M	19.0	9.0	90.0
4	Mary	25.0	F	5.0	7.0	63.0
5	Carli	NaN	F	0.0	6.0	15.0
6	Steve	15.0	M	7.0	4.0	46.0

df.dropna()



	Name	Age	Sex	Goals	Assists	Value
0	John	21.0	M	5.0	7.0	55.0
3	Wale	19.0	M	19.0	9.0	90.0
4	Mary	25.0	F	5.0	7.0	63.0
6	Steve	15.0	M	7.0	4.0	46.0

df.dropna(how=‘all')



	Name	Age	Sex	Goals	Assists	Value
0	John	21.0	M	5.0	7.0	55.0
1	Paul	23.0	NaN	10.0	4.0	84.0
3	Wale	19.0	M	19.0	9.0	90.0
4	Mary	25.0	F	5.0	7.0	63.0
5	Carli	NaN	F	0.0	6.0	15.0
6	Steve	15.0	M	7.0	4.0	46.0

Missing Value Treatment

- **Mean/ Mode/ Median Imputation:**

Imputation is a method to fill in the missing values with estimated ones.

The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Mean / Mode / Median imputation is one of the most frequently used methods.

It consists of replacing the missing data for a given attribute by the mean or median (quantitative attribute) or mode (qualitative attribute) of all known values of that variable. It can be of two types:-

Generalized Imputation: In this case, we calculate the mean or median for all non missing values of that variable then replace missing value with mean or median.

Similar case Imputation: We take the mean of non missing values of similar relevant column and impute the missing value with that.

Imputation Methods

The idea behind the imputation approach is to replace missing values with other sensible values. As you always lose information with the deletion approach when dropping either samples (rows) or entire features (columns), imputation is often the preferred approach. The many imputation techniques can be divided into two subgroups: *single imputation or multiple imputation*.

In **single imputation**, a single / one imputation value for each of the missing observations is generated. The imputed value is treated as the true value, ignoring the fact that no imputation method can provide the exact value. Therefore, single imputation does not reflect the uncertainty of the missing values.

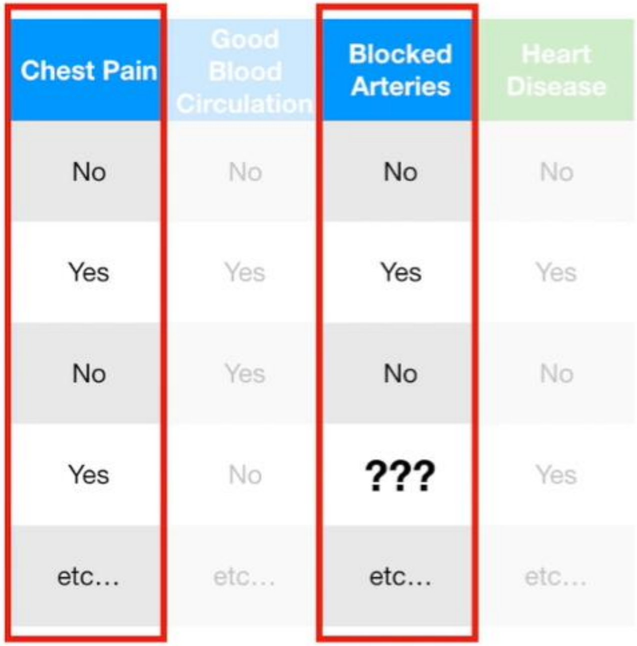
In **multiple imputation**, many imputed values for each of the missing observations are generated. This means many complete datasets with different imputed values are created. The analysis (e.g. training a linear regression to predict a target column) is performed on each of these datasets and the results are polled. Creating multiple imputations, as opposed to single imputations, accounts for the statistical uncertainty in the imputations.

Single Imputation

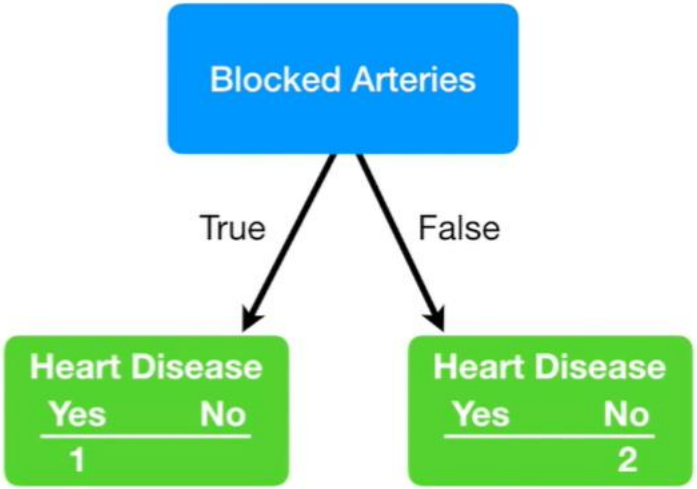
Most imputation methods are single imputation methods, following three main strategies: replacement by existing values, replacement by statistical values, and replacement by predicted values. Depending on the values used for each one of these strategies, we end up with methods that work on numerical values only and methods that work on both numerical and nominal columns.

Replacement by:	**Numerical Features Only**	**Numerical and Nominal Features**
Existing values	Minimum / Maximum	Previous / Next / Fixed
Statistical values	(Rounded) Mean / Median / Moving Average, Linear / Average Interpolation	Most Frequent
Predicted values	Regression Algorithms	Regression & Classification Algorithms, k-Nearest Neighbours

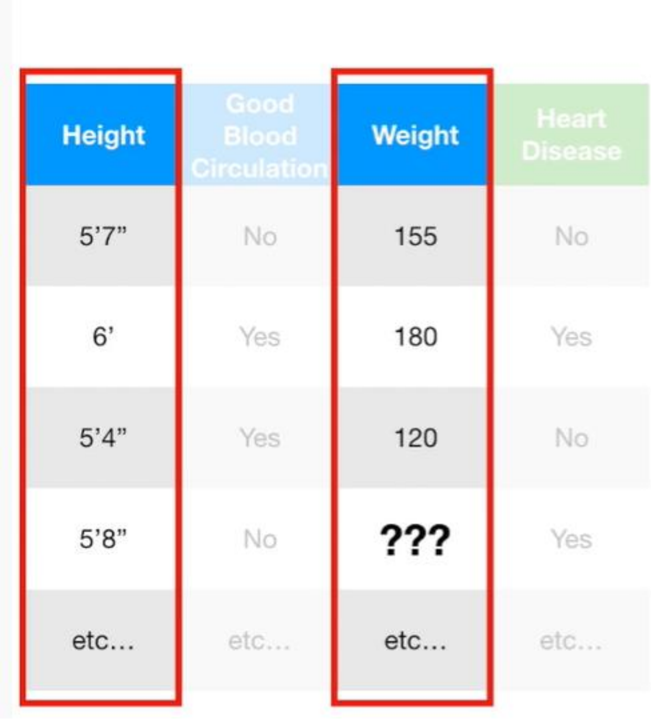
Alternatively, we could find another column that has the highest correlation with blocked arteries and use that as a guide.



Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
No	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

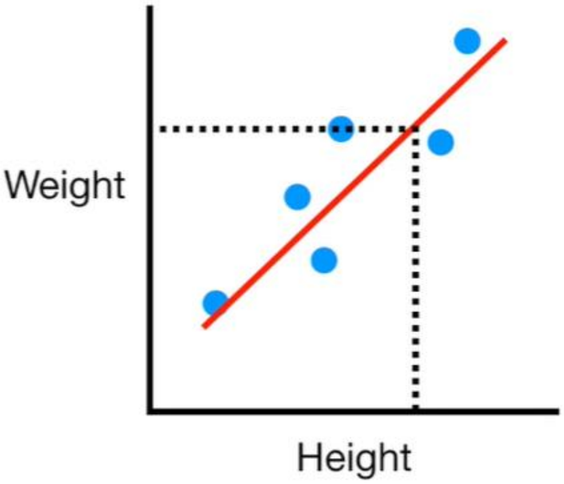


…and do a linear regression on the two columns…



Height	Good Blood Circulation	Weight	Heart Disease
5’7”	No	155	No
6’	Yes	180	Yes
5’4”	Yes	120	No
5’8”	No	???	Yes
etc…	etc…	etc…	etc…

…and use the least squares line to predict the value for weight.



(Rounded) Mean / Median Value / Moving Average

Other common imputation methods for numerical features are mean, rounded mean, or median imputation. In this case, the method substitutes the missing value with the mean, the rounded mean, or the median value calculated for that feature on the whole dataset. **In the case of a high number of outliers in your dataset, it is recommended to use the median instead of the mean.**

Most Frequent Value

Another common method that works for both numerical and nominal features uses the most frequent value in the column to replace the missing values.

Previous / Next Value

There are special imputation methods for time series or ordered data. These methods take into account the sorted nature of the dataset, where close values are probably more similar than distant values. A common approach for imputing missing values in time series substitutes the next or previous value to the missing value in the time series. This approach works for both numerical and nominal values.

1. **Impute with ZERO**
   - Impute missing values with zero
2. **Impute with Median/Mean/Mode**
   - For numerical variables, impute missing values with Mean or Median
   - For categorical variables, impute missing values with Mode
3. **Segment based imputation**
   - Identify relevant segments
   - Calculate mean/median/mode of segments
   - Impute the missing value according to the segments
   - For example, we can say rainfall hardly varies for cities in a particular State
   - In this case, we can impute missing rainfall value of a city with the average of that state

Fixed Value

Fixed value imputation is a general method that works for all data types and consists of substituting the missing value with a fixed value. The aggregated customer example uses fixed value imputation for numerical values. As an example of using fixed value imputation on nominal features, you can impute the missing values in a survey with “not answered”.

Minimum / Maximum Value

If you know that the data has to fit a given range [minimum, maximum], and if you know from the data collection process that the measuring system stops recording and the signal saturates beyond one of such boundaries, you can use the range minimum or maximum as the replacement value for missing values. For example, if in the monetary exchange a minimum price has been reached and the exchange process has been stopped, the missing monetary exchange price can be replaced with the minimum value of the law’s exchange boundary.

Linear / Average Interpolation

Similarly to the previous/next value imputation, but only applicable to numerical values, is linear or average interpolation, which is calculated between the previous and next available value, and substitutes the missing value. Of course, as for all operations on ordered data, it is important to sort the data correctly in advance, e.g. according to a timestamp in the case of time series data.

K Nearest Neighbors **(kNN)**

The idea here is to look for the k closest samples in the dataset where the value in the corresponding feature is not missing and to take the feature value occurring most frequently in the group as a replacement for the missing value.

Missing Value Prediction

Another common option for single imputation is to train a machine learning model to predict the imputation values for feature x based on the other features.

The rows without missing values in feature x are used as a training set and the model is trained based on the values in the other columns. Here we can use any classification or regression model, depending on the data type of the feature. After training, the model is applied to all samples with the feature missing value to predict its most likely value.

In the case of missing values in more than one feature column, all missing values are first temporarily imputed with a basic imputation method, e.g. the mean value. Then the values for one column are set back to missing. The model is then trained and applied to fill in the missing values. In this way, one model is trained for each feature with missing values, until all missing values are imputed by a model.

Prediction Model:

Prediction model is one of the sophisticated method for handling missing data. Here, we create a predictive model to estimate values that will substitute the missing data.

In this case, we divide our data set into two sets: One set with no missing values for the variable and another one with missing values. First data set become training data set of the model while second data set with missing values is test data set and variable with missing values is treated as target variable.

Next, we create a model to predict target variable based on other attributes of the training data set and populate missing values of test data set.
We can use regression, ANOVA, Logistic regression and various modelling technique to perform this.

There are 2 drawbacks for this approach:

- The model estimated values are usually more well-behaved than the true values.
- If there are no relationships with attributes in the data set and the attribute with missing values, then the model will not be precise for estimating missing values.

KNN (k-Nearest Neighbours) Imputation:

In this **method**, k neighbors are chosen based on some distance measure (Euclidean distance) and their average is used as an **imputation** estimate. **KNN** can predict both discrete (most frequent value among the k nearest neighbors) and continuous attributes (mean among the k nearest neighbors)

Advantages:

- k-nearest neighbour can predict both qualitative & quantitative attributes.
- Creation of predictive model for each attribute with missing data is not required.
- Attributes with multiple missing values can be easily treated
- Correlation structure of the data is taken into consideration

Disadvantages:

- KNN algorithm is very time-consuming in analyzing large database. It searches through all the dataset looking for the most similar instances.
- Choice of k-value is very critical. Higher value of k would include attributes which are significantly different from what we need whereas lower value of k implies missing out of significant attributes

Multiple imputation- MICE --Iterative Imputer

Multiple Imputation by Chained Equations (MICE)

Multiple Imputation by Chained Equations (MICE) is a robust, informative method for dealing with missing values in datasets. MICE operates under the assumption that the missing data are Missing At Random (MAR) or Missing Completely At Random (MCAR).

The procedure is an extension of the single imputation procedure by “Missing Value Prediction” this is step 1. However, there are two additional steps in the MICE procedure.

Step 1: This is the process as in the imputation procedure by “Missing Value Prediction” on a subset of the original data. One model is trained to predict the missing values in one feature, using the other features in the data row as the independent variables for the model. This step is repeated for all features. This is a cycle or iteration.

Step 2: Step 1 is repeated k times, each time using the most recent imputations for the independent variables, until convergence is reached. Most often, k=10 cycles are sufficient.

Step 3: The whole process is repeated N times on N different random subsets. The resulting N models will be slightly different, and will produce N slightly different predictions for each missing value.

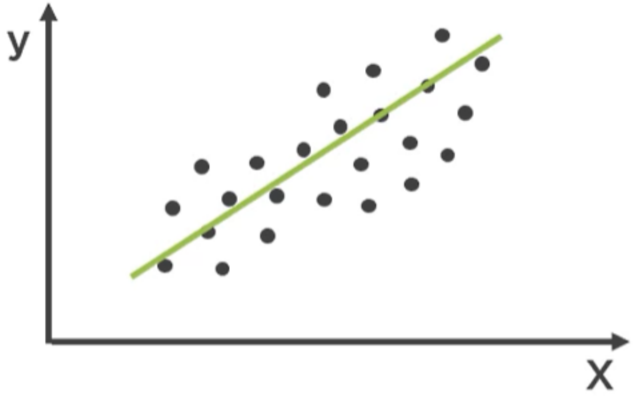
The analysis, e.g. training a linear regression for a target variable, is now performed on each one of the N final datasets. Finally the results are combined, often this is also called pooling.

This provides more robust results than by single imputation alone. Of course, the downside of such robustness is the increase in computational complexity.

Multiple Imputation by Chained Equations

Age	Income	Gender
33	$40-60K	M
18	$40-60K	M
15	$60-80K	F
35.3	$40-60K	F

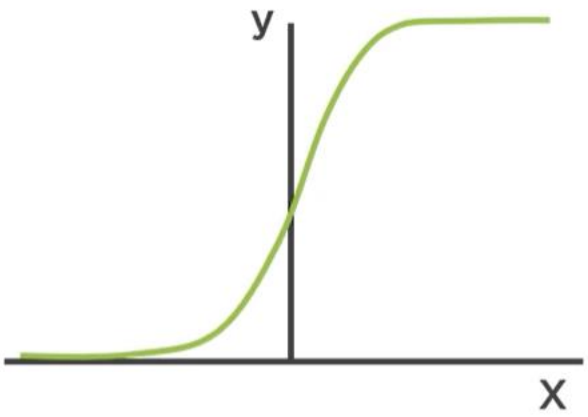
Age ~ f(Income, Gender)



Multiple Imputation by Chained Equations

Age	Income	Gender
33	?	M
18	$40-60K	M
15	$60-80K	F
35.3	$40-60K	F

Income ~ f(Age, Gender)



Multiple Imputation by Chained Equations

Age	Income	Gender
33	$40-60K	M
18	$40-60K	M
15	$60-80K	F
35.3	$40-60K	F

Age	Income	Gender
33	$40-60K	M
18	$40-60K	M
15	$60-80K	F
34.2	$40-60K	F

Age	Income	Gender
33	$60-80K	M
18	$40-60K	M
15	$60-80K	F
34.0	$40-60K	F

Age	Income	Gender
33	$60-80K	M
18	$40-60K	M
15	$60-80K	F
33.8	$40-60K	F

n=4

MICE Steps

The chained equation process can be broken down into four general steps:

Step 1: A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as “place holders.”

Step 2: The “place holder” mean imputations for one variable (“var”) are set back to missing.

Step 3: The observed values from the variable “var” in Step 2 are regressed on the other variables in the imputation model, which may or may not consist of all of the variables in the dataset. In other words, “var” is the dependent variable in a regression model and all the other variables are independent variables in the regression model. These regression models operate under the same assumptions that one would make when performing linear, logistic, or Poison regression models outside of the context of imputing missing data.

Step 4: The missing values for “var” are then replaced with predictions (imputations) from the regression model. When “var” is subsequently used as an independent variable in the regression models for other variables, both the observed and these imputed values will be used.

Step 5: Steps 2–4 are then repeated for each variable that has missing data. The cycling through each of the variables constitutes one iteration or “cycle.” At the end of one cycle all of the missing values have been replaced with predictions from regressions that reflect the relationships observed in the data.

Step 6: Steps 2–4 are repeated for a number of cycles, with the imputations being updated at each cycle.

The number of cycles to be performed can be specified by the researcher. At the end of these cycles the final imputations are retained, resulting in one imputed dataset. Generally, 10 cycles are performed (Raghunathan *et al.*, 2002); however, research is needed to identify the optimal number of cycles when imputing data under different conditions.

To make the chained equation approach more concrete, imagine a simple example where we have three variables in our dataset: **age, income, and gender,** and all three have at least some missing values.

The MAR assumption would imply that the probability of a particular variable being missing depends only on the observed values, and that, for example, whether someone's income is missing does not depend on their (unobserved) income.

In Step 1 of the MICE process, each variable would first be imputed using, e.g. mean imputation, temporarily setting any missing value equal to the mean observed value for that variable

Then in Step 2 the imputed mean values of age would be set back to missing.

In Step 3, a linear regression of age predicted by income and gender would be run using all cases where age was observed. In Step 4, predictions of the missing age values would be obtained from that regression equation and imputed. At this point, age does not have any missing ness.

Steps 2–4 would then be repeated for the income variable. The originally missing values of income would be set back to missing and a linear regression of income predicted by age and gender would be run using all cases with income observed; imputations (predictions) would be obtained from that regression equation for the missing income values.

Then, Steps 2–4 would again be repeated for the variable gender. The originally missing values of gender would be set back to missing and a logistic regression of gender on age and income would be run using all cases with gender observed; predictions from that logistic regression model would be used to impute the missing gender values.

This entire process of iterating through the three variables would be repeated until convergence; the observed data and the final set of imputed values would then constitute one “complete” data set.

How to choose the correct strategy

Two common approaches to imputing missing values is to replace all missing values with either a fixed value, for example zero, or with the mean of all available values. Which approach is better?

Let’s see the effects on two different case studies:

Case Study 1: threshold-based anomaly detection on sensor data
Case Study 2: a report of customer aggregated data

Case Study 1: Imputation for threshold-based anomaly detection

In a classic threshold-based solution for anomaly detection, a threshold, calculated from the mean and variance of the original data, is applied to the sensor data to generate an alarm. If the missing values are imputed with a fixed value, e.g. zero, this will affect the calculation of the mean and variance used for the threshold definition. This would likely lead to a wrong estimate of the alarm threshold and to some expensive downtime.

Here imputing the missing values with the mean of the available values is the right way to go.

Case Study 2: Imputation for aggregated customer data

In a classic reporting exercise on customer data, the number of customers and the total revenue for each geographical area of the business needs to be aggregated and visualized, for example via bar charts. The customer dataset has missing values for those areas where the business has not started or has not picked up and no customers and no business have been recorded yet. In this case, using the mean value of the available numbers to impute the missing values would make up customers and revenues where neither customers nor revenues are present.

The right way to go here is to impute the missing values with a fixed value of zero.

In both cases, it is our knowledge of the process that suggests to us the right way to proceed in imputing missing values. In the case of sensor data, missing values are due to a malfunctioning of the measuring machine and therefore real numerical values are just not recorded. In the case of the customer dataset, missing values appear where there is nothing to measure yet.

You see already from these two examples, that there is no panacea for all missing value imputation problems and clearly we can’t provide an answer to the classic question: “which strategy is correct for missing value imputation for my dataset?” The answer is too dependent on the domain and the business knowledge.

Take Away :

Summarizing, we can reach the following conclusions.

Use listwise deletion (“deletion”) carefully, especially on small datasets. When removing data, you are removing information. Not all datasets have redundant information to spare!

When using fixed value imputation, you need to know what that fixed value means in the data domain and in the business problem. Here, you are injecting arbitrary information into the data, which can bias the predictions of the final model.

If you want to impute missing values without prior knowledge it is hard to say which imputation method works best, as it is heavily dependent on the data itself.

In the end, nothing beats prior knowledge of the task and of the data collection process!

Fixing Rows & Columns

Fixing Rows

- Delete summary rows; Delete incorrect rows: Header rows, Footer rows
- Delete extra rows: Blank rows, Page No.etc.

Fixing Columns

- Delete columns: Delete unrequired columns
- Rename columns consistently: Abbreviations, encoded columns
- Add column names: Add column names if missing in dataset
- Merge columns for creating unique identifiers if needed: E.g. Merge Country, State, District, City into Full address
- Split columns for more data: Split address to get Country, City, District & State etc to analyse each separately(derived columns)
- Align misaligned columns

Standardizing Values

Scale values if required: Make sure the observations under a variable have a same scale.

Standardize units: Ensure all observations under a variable have a common and consistent unit, e.g. convert miles/hr to km/hr, lbs to kgs, etc.

Standardize precision for better presentation of data, e.g. 9.818181 kgs to 9.82 kgs.

Remove outliers: Remove high and low values that would disproportionately affect the results of your analysis.

Fixing invalid values

Encode-Decode properly: In case the data is being read as junk characters, try to change encoding, E.g. CP1252 instead of UTF-8.

Correct values go beyond range: If some of the values are beyond logical range, e.g. temperature less than -273° C (0° K), you would need to correct them as required.

Convert incorrect data types: Correct the incorrect data types to the correct data types for ease of analysis. E.g. if numeric values are stored as strings, it would not be possible to calculate metrics such as mean, median, etc. Some of the common data type corrections are — number to string: “PIN Code 560009” to “560009"; string to number: “24,300" to “24300”; string to date: "2019-Aug" to “2019/08”; etc.

Correct values not in the list: Remove values that don't belong to a list. E.g. In a data set containing blood groups of individuals as "E" or "C" are invalid values and can be removed.

Correct wrong structure: Values that don't follow a defined structure can be removed. E.g. In a data set containing pin codes of Indian cities, a pin code of 10 digits would be an invalid value and needs to be removed. Similarly, a phone number of 9 or 11 digits would be an invalid value.

Validate internal rules: If there are internal rules such as a date of a product's delivery must definitely be after the date of the order, they should be correct and consistent.

Filtering Data

After you have fixed the missing values, standardized the existing values, and fixed the invalid values, you would get to the last stage of data cleaning. Though you have a largely accurate data set by now, you might not need the entire data set for your analysis.

It is important to understand what you need to infer from the data and then choose the relevant parts of the data set for your analysis. Thus, you need to filter the data to get what you need for your analysis.

Filter rows: Filter by segment, filter by date period to get only the rows relevant to the analysis
Filter columns: Pick columns relevant to the analysis
Aggregate data: Group by required keys, aggregate the rest
De-duplicate data: Remove identical rows, remove rows where some columns are identical.

Outliers

Outliers are observations in a dataset that don’t fit in some way.

Perhaps the most common or familiar type of outlier is the observations that are far from the rest of the observations or the center of mass of observations.

This is easy to understand when we have one or two variables and we can visualize the data as a histogram or scatter plot, although it becomes very challenging when we have many input variables defining a high-dimensional input feature space.

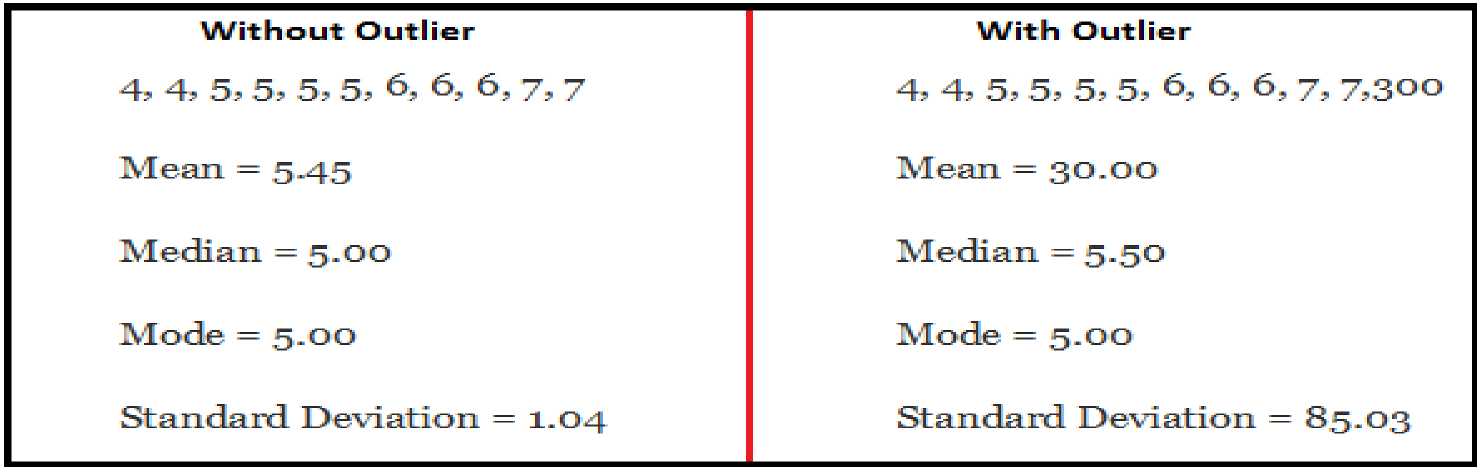
In this case, simple statistical methods for identifying outliers can break down, such as methods that use standard deviations or the interquartile range.

It can be important to identify and remove outliers from data when training machine learning algorithms for predictive modeling.

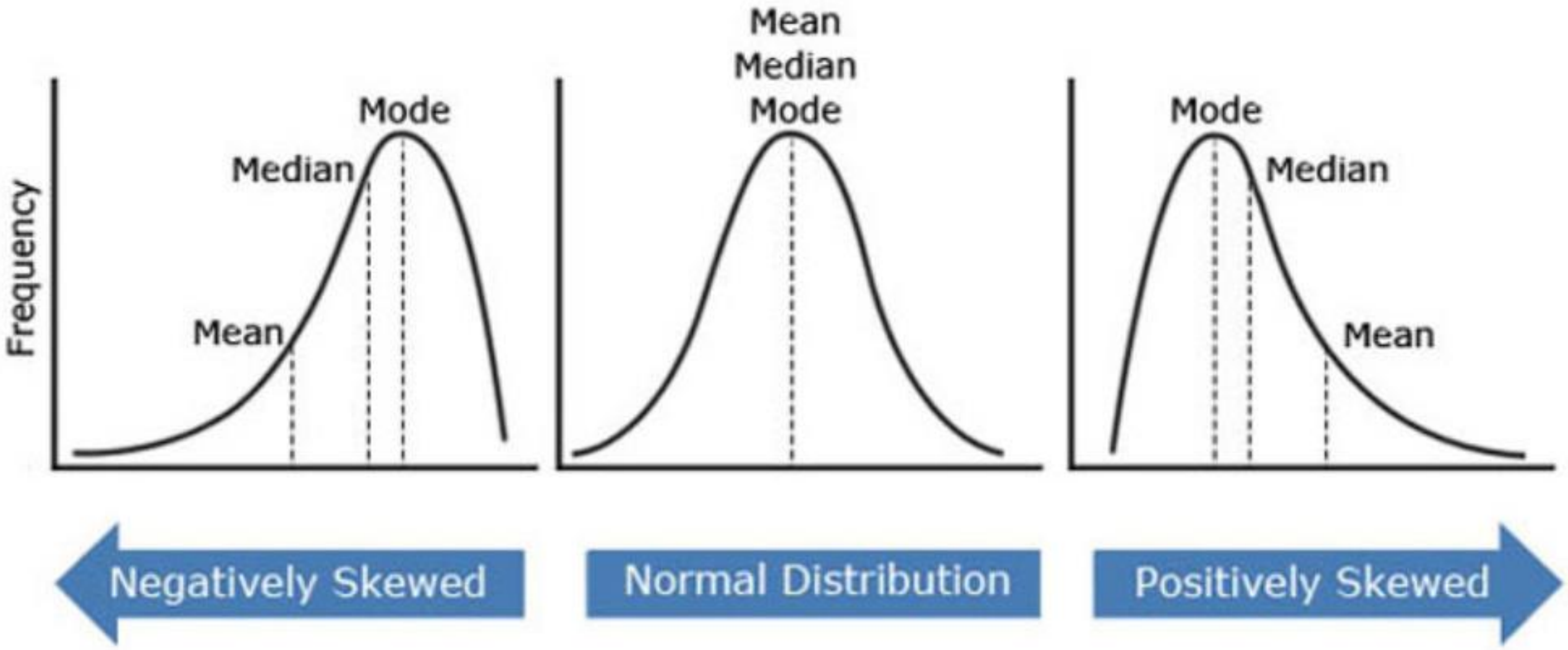
Outliers can skew statistical measures and data distributions, providing a misleading representation of the underlying data and relationships. Removing outliers from training data prior to modeling can result in a better fit of the data and, in turn, more skillful predictions.

What is an Outlier?

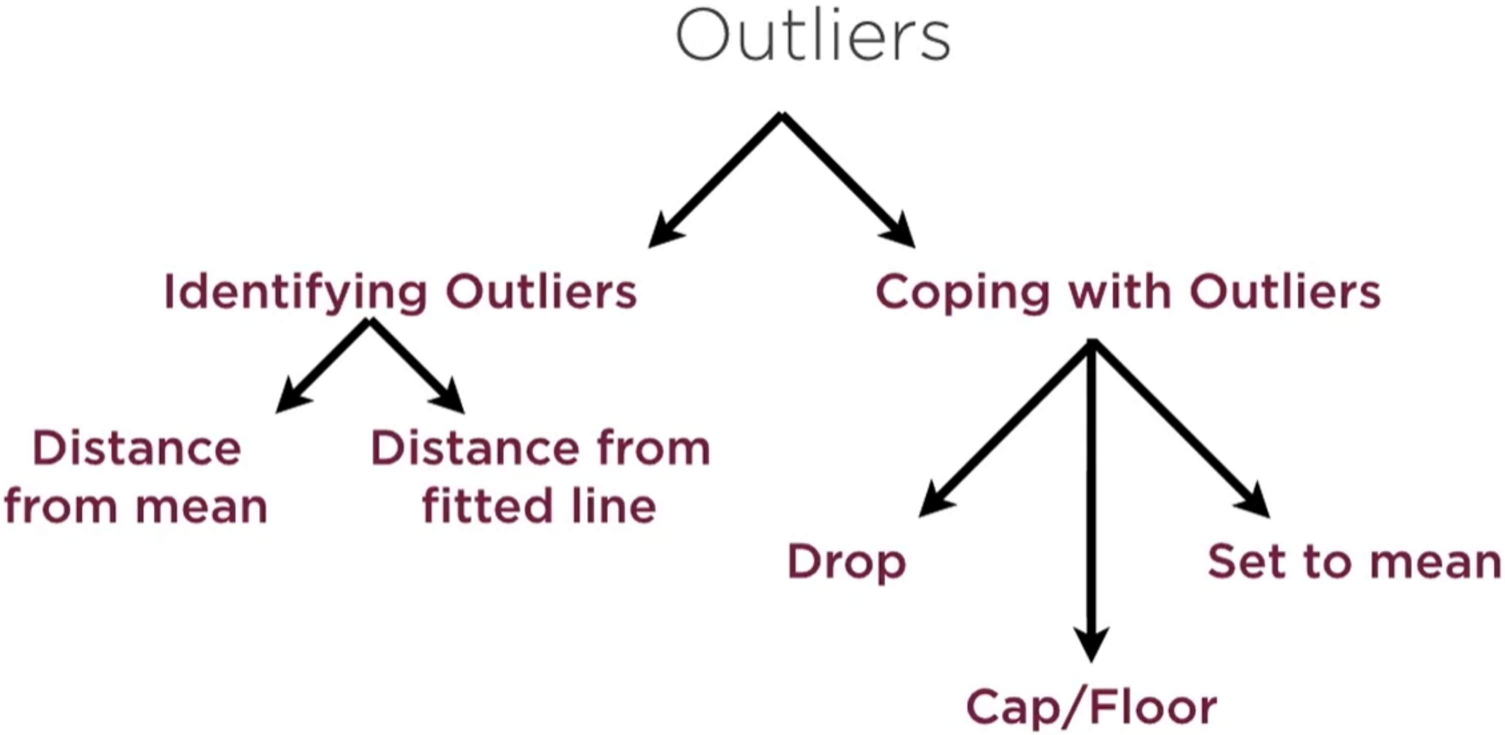
Outlier is a commonly used terminology by analysts and data scientists as it needs close attention else it can result in wildly wrong estimations. Simply speaking, Outlier is an observation that appears far away and diverges from an overall pattern in a sample.



Without Outlier	With Outlier
4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7	4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7,300
Mean = 5.45	Mean = 30.00
Median = 5.00	Median = 5.50
Mode = 5.00	Mode = 5.00
Standard Deviation = 1.04	Standard Deviation = 85.03



skewness is a measure of the asymmetry of the probability distribution of a real valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.



Outliers

Identifying Outliers

Distance from mean

Distance from fitted line

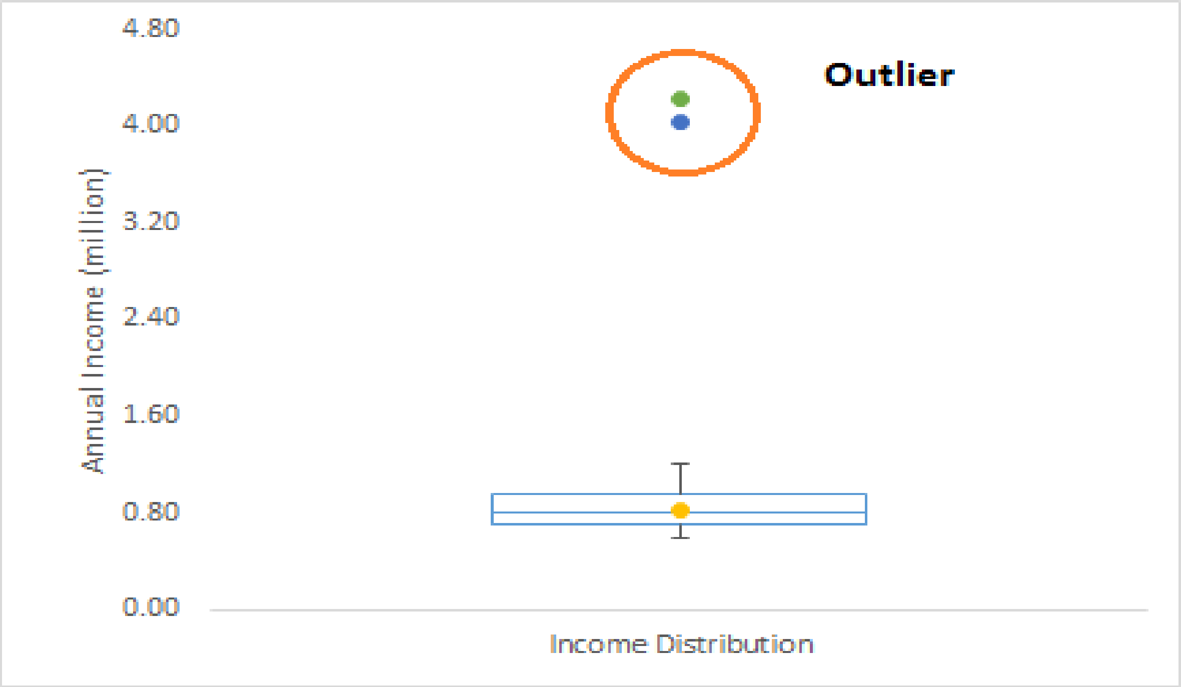
Coping with Outliers

Drop

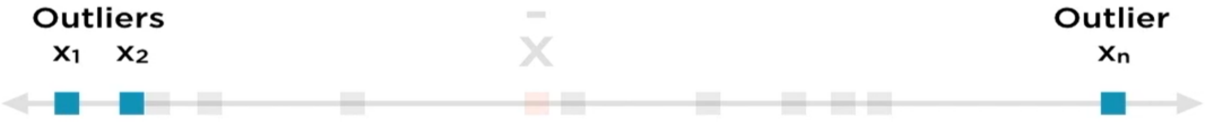
Cap/Floor

Set to mean

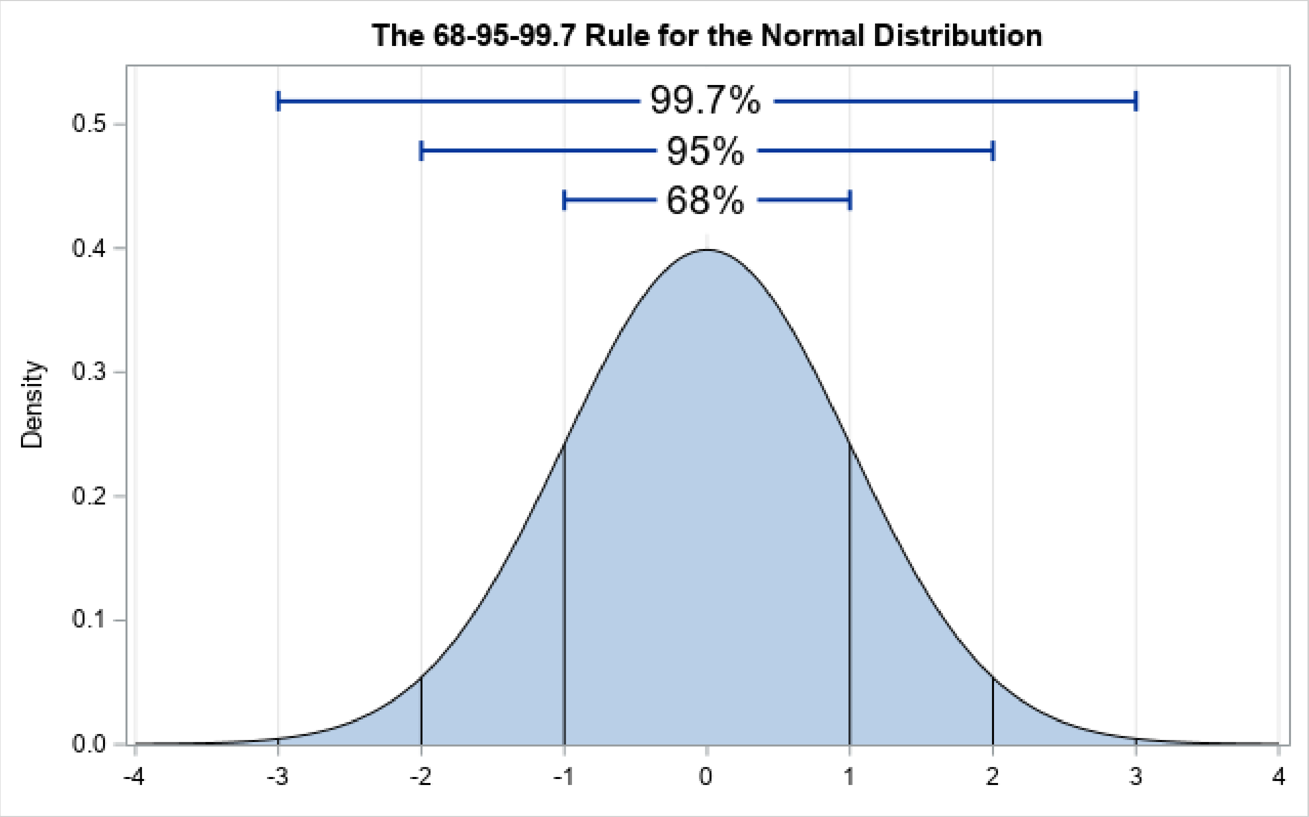
Most commonly used method to **detect outliers** is visualization. We use various visualization methods, like Box-plot, Histogram, Scatter Plot.



Outliers



Points that lie more than 3 standard deviations from the mean are often considered outliers



What is the impact of Outliers on a dataset?

Outliers can drastically change the results of the data analysis and statistical modelling.

There are numerous unfavourable impacts of outliers in the data set:

- It increases the error variance and reduces the power of statistical tests
- If the outliers are non-randomly distributed, they can decrease normality.
- They can bias or influence estimates that may be of substantive interest.
- They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

- **Different applications may have very different requirements.**

- For example, **in clinic data analysis, a small deviation may be important enough to justify an outlier.**

- **In contrast, in marketing analysis, objects are often subject to larger fluctuations, and consequently a substantially larger deviation is needed to justify an outlier.**

- Outlier detection's high dependency on the application type, makes it impossible to develop a universally applicable outlier detection method.

- Instead, individual outlier detection methods that are dedicated to specific applications must be developed.

While in many contexts outliers are considered as noise that must be eliminated, as pointed out elsewhere, **“one person’s noise could be another person’s signal”,** and thus outliers themselves can be of great interest.

- Outlier mining is used in telecom or credit card frauds to detect the atypical usage of telecom services or credit cards, in intrusion detection for detecting unauthorized accesses, in medical analysis to test abnormal reactions to new medical therapies, in marketing and customer segmentations to identify customers spending much more or much less than average customer, in surveillance systems, in data cleaning, and in many other fields.

- Application-specific outlier detection : Technically, choosing the similarity/distance measure and the relationship model to describe data objects is critical in outlier detection. Unfortunately, such are often application-dependent.

Handling noise in outlier detection : Outliers are different from noise. It is also well known that the quality of real data sets tends to be poor. Noise often unavoidably exists in data collected in many applications. Noise may be present as deviations in attribute values or even as missing values. Low data quality and the presence of noise bring a huge challenge to outlier detection. They can distort the data, blurring the distinction between normal objects and outliers. Moreover, noise and missing data may “hide” outliers and reduce the effectiveness of outlier detection—an outlier may as a noise point, and an outlier detection method may mistakenly identify a noise point as an outlier.

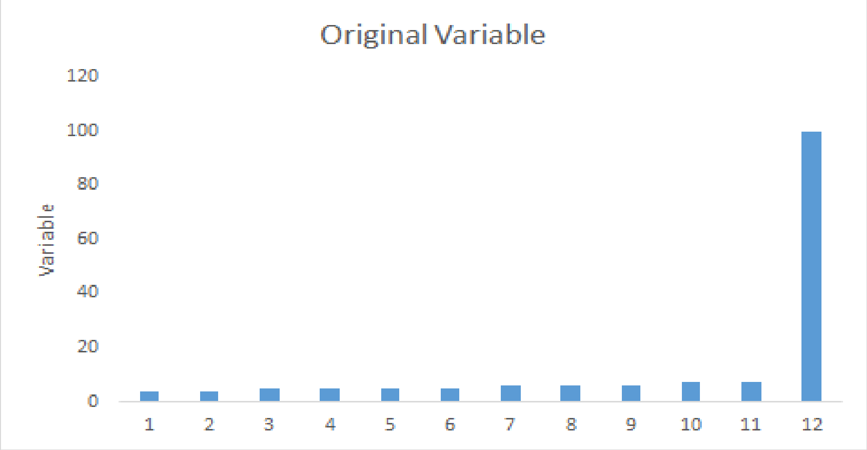
Understandability : In some application scenarios, a user may want to not only detect outliers, but also understand why the detected objects are outliers.

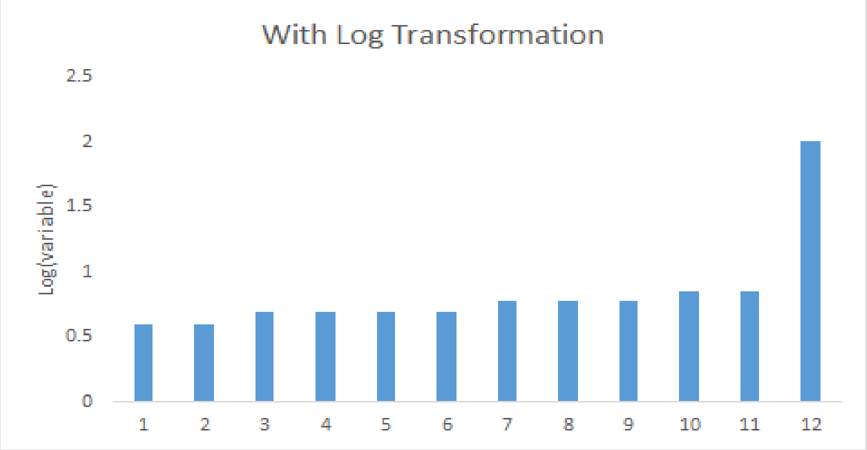
To meet the understandability requirement, an outlier detection method has to provide some justification of the detection. For example, a statistical method can be used to justify the degree to which an object may be an outlier based on the likelihood that the object was generated by the same mechanism that generated the majority of the data.

The smaller the likelihood, the more unlikely the object was generated by the same mechanism, and the more likely the object is an outlier.

Symmetric distribution is preferred over skewed distribution as it is easier to interpret and generate inferences. Some modelling techniques requires normal distribution of variables. So, whenever we have a skewed distribution, we can use transformations which reduce skewness.

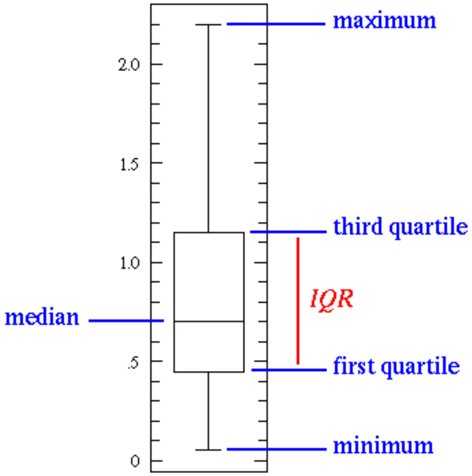
For right skewed distribution, we take square / cube root or logarithm of variable and for left skewed, we take square / cube or exponential of variables.





How to detect Outliers?

Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot**, **Histogram**, **Scatter Plot** (above, we have used box plot and scatter plot for visualization). Some analysts also various thumb rules to detect outliers.



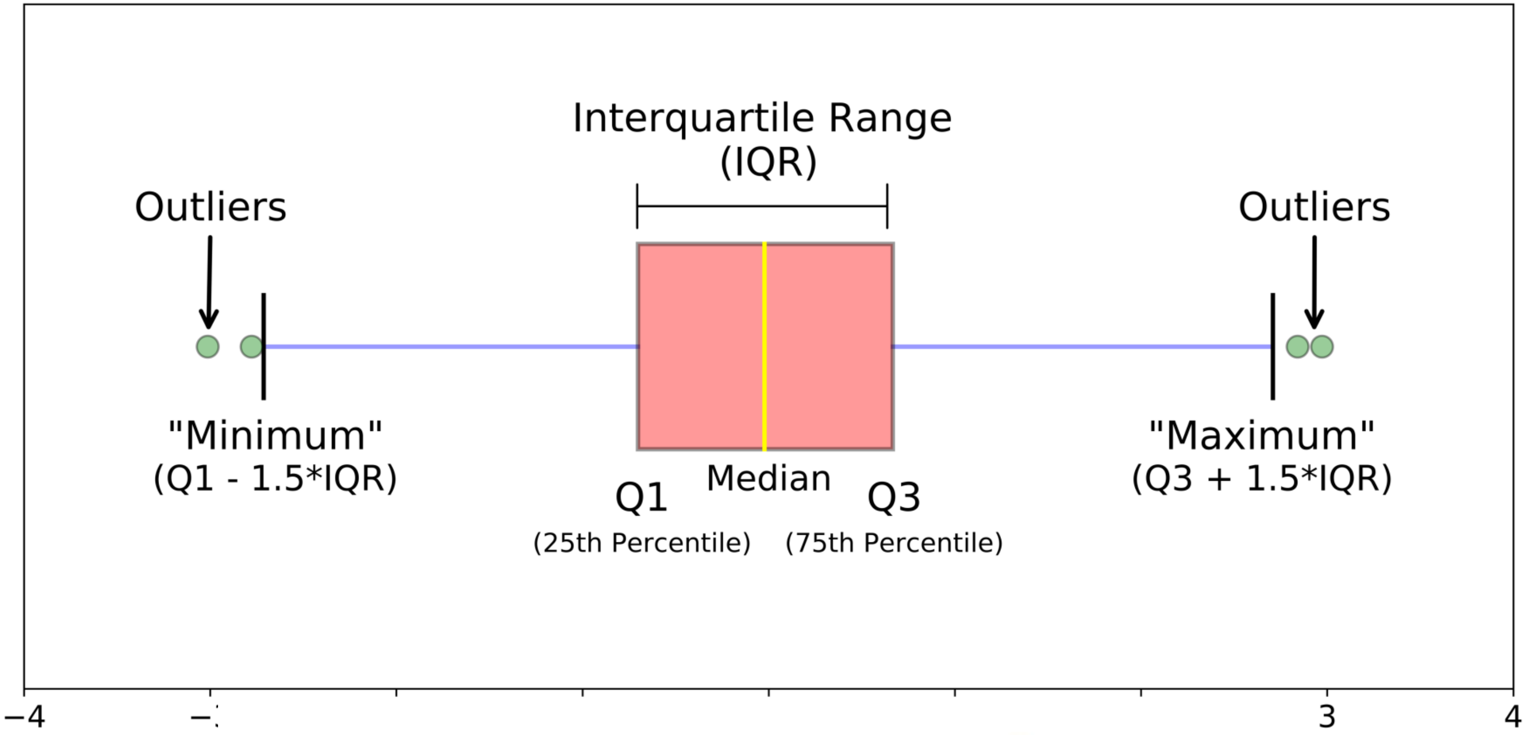
Some of them are:
Any value, which is beyond the range of
(Q1-1.5 x IQR   to   Q3+1.5 x IQR)

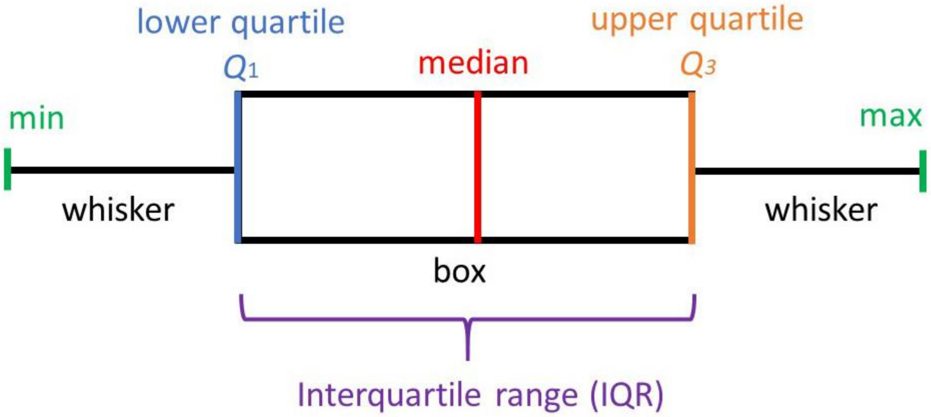
Inter quantile range IQR=(Q3-Q1)

Use capping methods. Any value which out of range of
5th and 95th percentile can be considered as outlier.

Data points, three or more standard deviation away from mean are considered outlier.

Outlier detection is merely a special case of the examination of data for influential data points and it also depends on the business understanding.


Interquartile Range
(IQR)
Outliers
Outliers
"Minimum"
(Q1 - 1.5*IQR)
Q1
(25th Percentile)
Median
Q3
(75th Percentile)
"Maximum"
(Q3 + 1.5*IQR)
−4
−
3
4


lower quartile
Q1
median
upper quartile
Q3
min
max
whisker
whisker
box
Interquartile range (IQR)

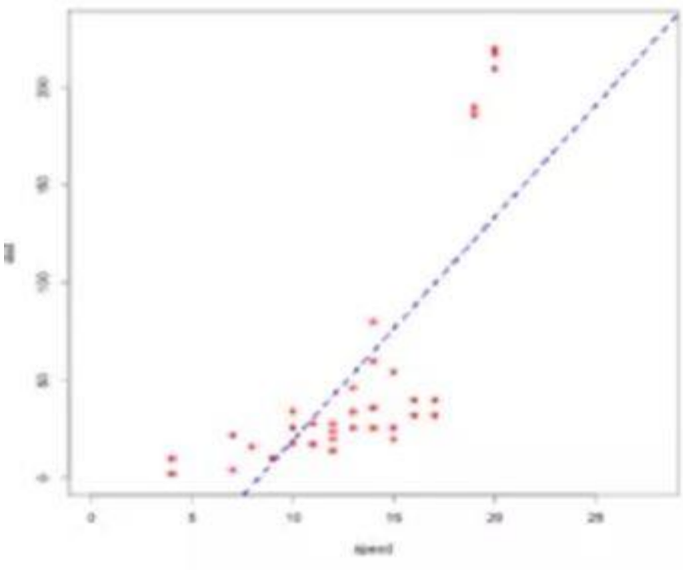
Use common sense

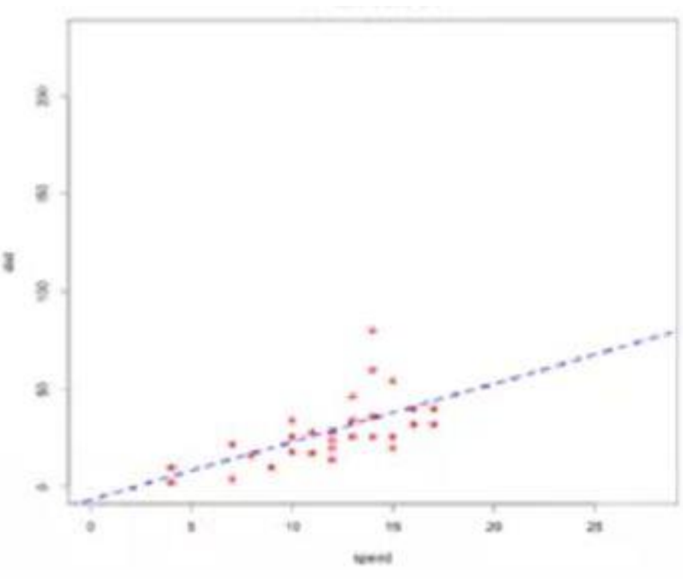
With data where you already know the distribution (like people's ages), you can use common sense to find outliers that were incorrectly recorded.

For example, you know that 345 is not a valid age, while 45 is.

When & Why Outliers Removal is needed ?

We should not drop any observations without careful consideration, as such information can be valuable to understand the unusual behavior or anomalies in the data unless we are sure that the outlier may be due to measurement error or if the presence of outliers influences the fit of a model.





How to remove Outliers?

Most of the ways to deal with outliers are similar to the methods of missing values like deleting observations, transforming them, binning them, treat them as a separate group, imputing values and other statistical methods.

Deleting observations: We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers. We can also use trimming at both ends to remove outliers.

Transforming and binning values: Transforming variables can also eliminate outliers. Natural log of a value reduces the variation caused by extreme values. Binning is also a form of variable transformation. Decision Tree algorithm allows to deal with outliers well due to binning of variable. We can also use the process of assigning weights to different observations.

Imputing: Like imputing missing values, we can also impute outliers. We can use mean, median, mode imputation methods. Before imputing values, we should analyse if it is natural outlier or artificial. If it is artificial, we can go with imputing values. We can also use statistical model to predict values of outlier observation and after that we can impute it with predicted values.

Treat separately: If there are significant number of outliers, we should treat them separately in the statistical model. One of the approach is to treat both groups as two different groups and build individual model for both groups and then combine the output.

Winsorising or Clamp Transformation: (Capping)

Winsorising replaces the outliers with the nearest nonsuspect data. This is also called clamp transformation as we clamp all values above an upper threshold and below a lower threshold to these threshold values, thus capping the values of outliers:

$$f_i = \begin{cases} lower, & if\ f_i < lower \\ upper, & if\ f_i > lower \\ f_i, & otherwise \end{cases}$$

where *fi* is a specific value of feature *f*, and lower and upper are the lower and upper thresholds., given by either the IQR method or z-score.

It is recommended to apply the clamp transformation in cases where it is suspected that a model is performing poorly due to the presence of outliers.

The better way to evaluate the impact of winsorising is by comparing the performance of different models trained on datasets where the transformation has been applied and where it has not.

Use Algorithms Robust to Outliers

Tree-based algorithms and boosting methods are insensitive to outliers due to the intrinsic nature of the recursive binary splitting approach used to partition the feature space. These algorithms are the best bet when there are outliers in the input features.

If there are outliers in the target variable, tree-based algorithms are good but care must be taken to choose the loss function. Reason being that if we use the mean squared error function, then the difference is squared and would highly influence the next tree since boosting attempts to fit the (gradient of the) loss. However, there are more robust error functions that can be used for boosted tree methods like Huber Loss and Mean Absolute Deviation loss.

Feature Engineering

Feature Engineering

Feature engineering is the science (and art) of extracting more information from existing data. You are not adding any new data here, but you are actually making the data you already have more useful.

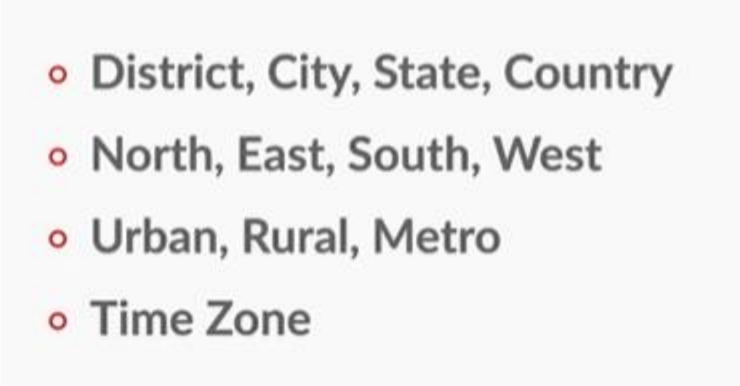
For example, let’s say you are trying to predict foot fall in a shopping mall based on dates. If you try and use the dates directly, you may not be able to extract meaningful insights from the data. This is because the foot fall is less affected by the day of the month than it is by the day of the week.

Now this information about day of week is implicit in your data. You need to bring it out to make your model better.

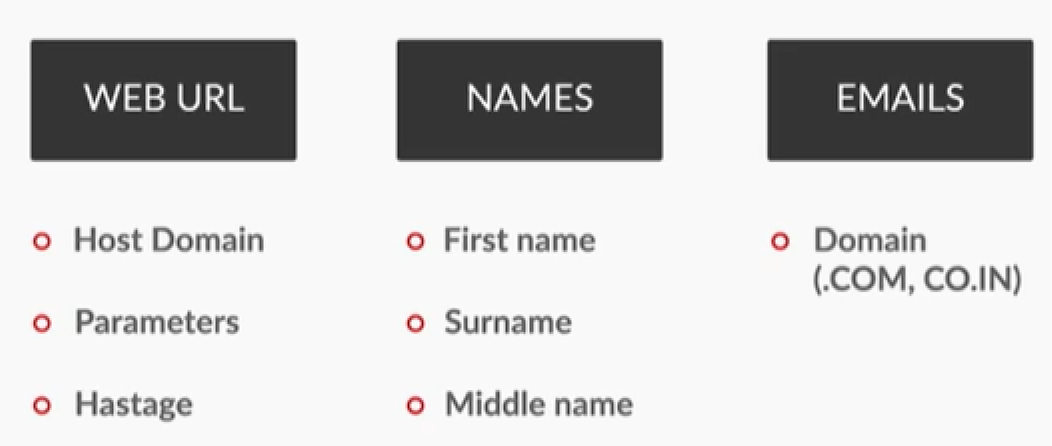
Feature engineering itself can be divided in 2 steps:

- Variable transformation.
- Variable / Feature creation
  - Business Driven Derived Variables
  - Data Driven Derived Variables
  - Type Driven Derived Variables

Type Driven Derived Variables



- District, City, State, Country
- North, East, South, West
- Urban, Rural, Metro
- Time Zone



WEB URL

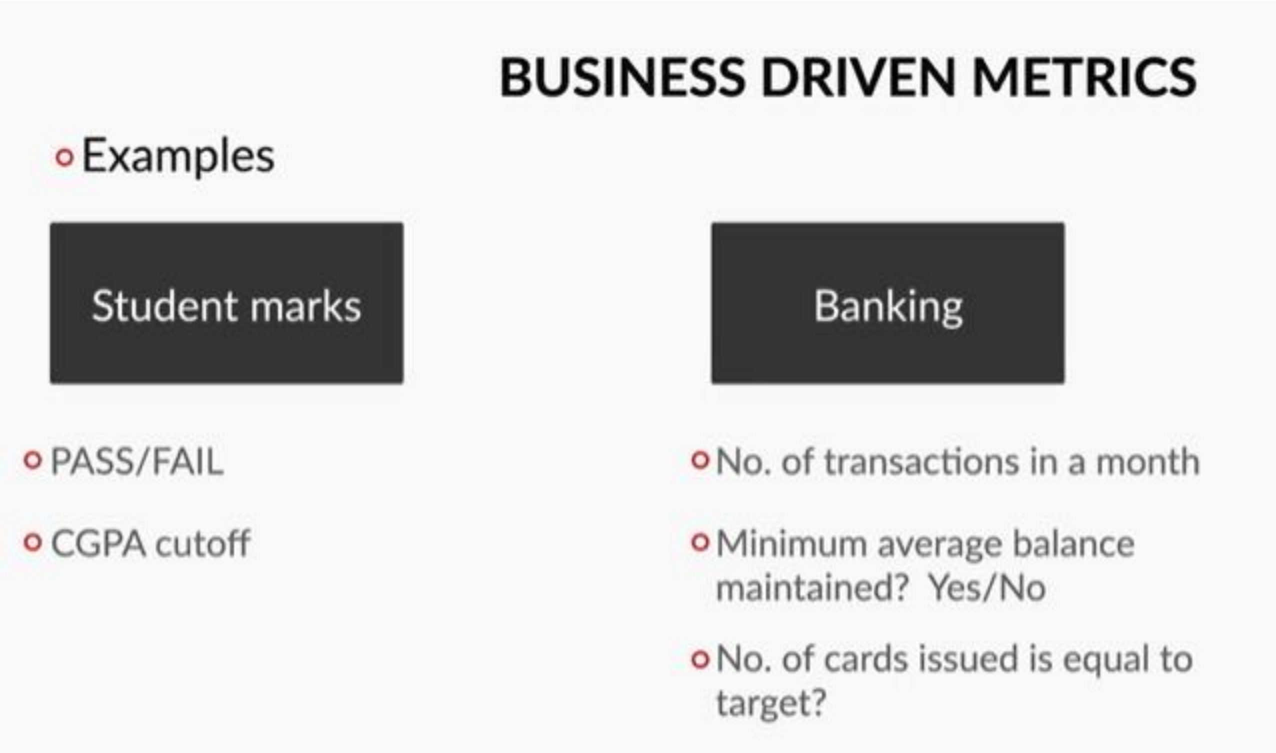
- Host Domain
- Parameters
- Hastage

NAMES

- First name
- Surname
- Middle name

EMAILS

- Domain (.COM, CO.IN)



BUSINESS DRIVEN METRICS

- Examples

Student marks

- PASS/FAIL
- CGPA cutoff

Banking

- No. of transactions in a month
- Minimum average balance maintained? Yes/No
- No. of cards issued is equal to target?

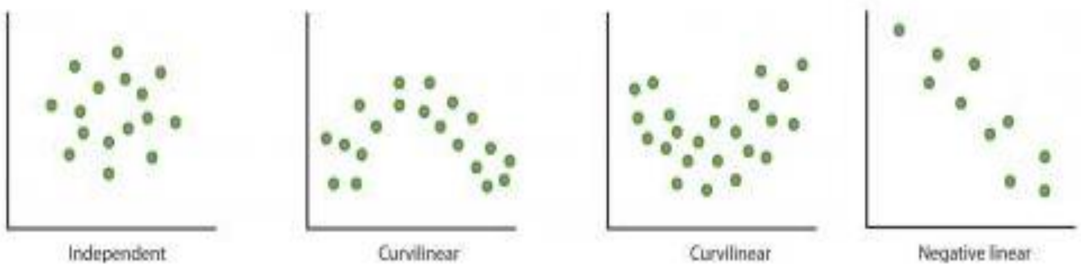
Non Usable Variables

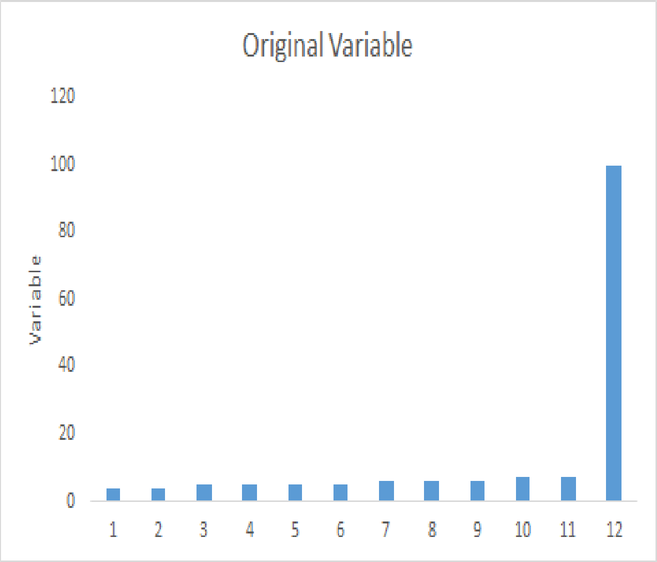
1. Variables with single unique value
2. Variables with low fill rate
3. Variables with regulatory issue
4. Variable with no business sense

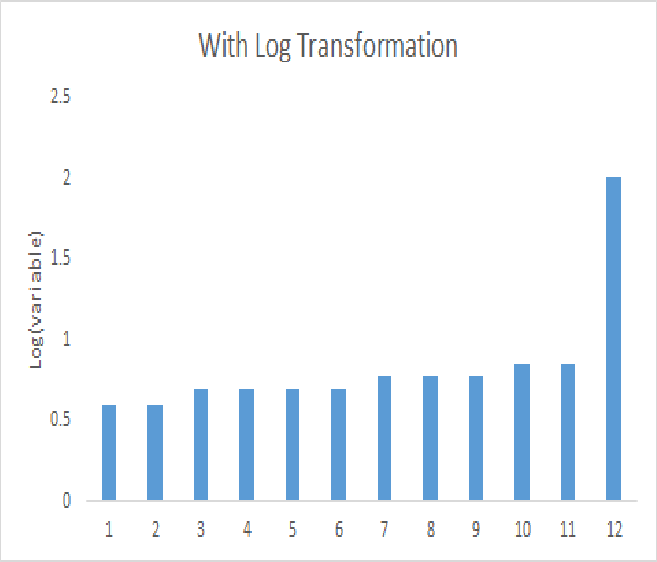
When should we use Variable Transformation?

Below are the situations where variable transformation is a requisite:

- When we want to **change the scale** of a variable or standardize the values of a variable for better understanding. While this transformation is a must if you have data in different scales, this transformation does not change the shape of the variable distribution.
- When we can **transform complex non-linear relationships into linear relationships**. Existence of a linear relationship between variables is easier to comprehend compared to a non-linear or curved relation. Transformation helps us to convert a non-linear relation into linear relation. Scatter plot can be used to find the relationship between two continuous variables. These transformations also improve the prediction. **Log transformation** is one of the commonly used transformation technique used in these situations.





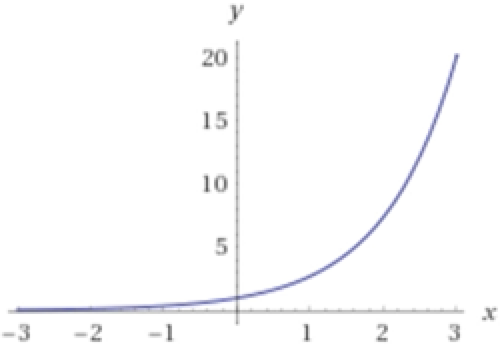


Variable Transformation is also done from an **implementation point of view** (Human involvement). Let’s understand it more clearly.

In one of my project on employee performance, I found that age has direct correlation with performance of the employee i.e. higher the age, better the performance.

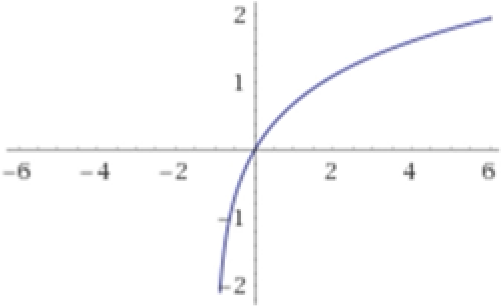
From an implementation stand point, launching age based programme might present implementation challenge. However, categorizing the sales agents in three age group buckets of <30 years, 30-45 years and >45 and then formulating three different strategies for each group is a judicious approach. This categorization technique is known as **Binning of Variables.**

If



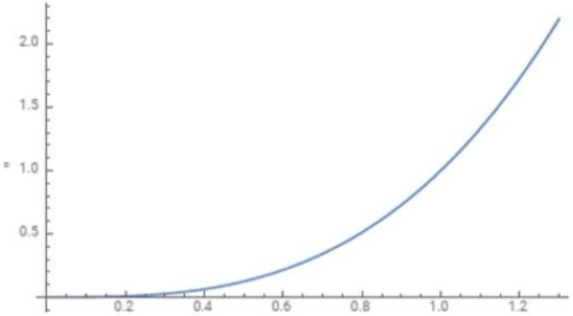
→ Take e^x instead of x

If



→ Take log(1+x) instead of x

If



→ Take x^2 *or* x^n instead of x

What are the common methods of Variable Transformation?

There are various methods used to transform variables. As discussed, some of them include square root, cube root, logarithmic, binning, reciprocal and many others. Let’s look at these methods in detail by highlighting the pros and cons of these transformation methods.

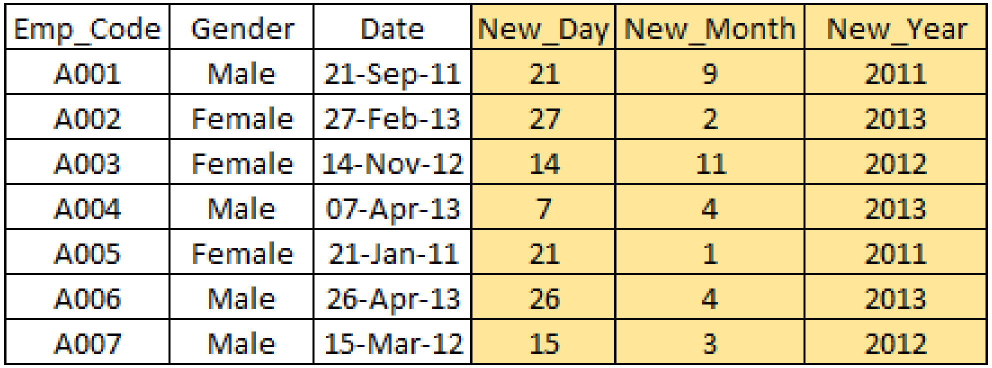
Logarithm: Log of a variable is a common transformation method used to change the shape of distribution of the variable on a distribution plot. It is generally used for reducing right skewness of variables. Though, It can’t be applied to zero or negative values as well.

Square / Cube root: The square and cube root of a variable has a sound effect on variable distribution. However, it is not as significant as logarithmic transformation. Cube root has its own advantage. It can be applied to negative values including zero. Square root can be applied to positive values including zero.

Binning: It is used to categorize variables. It is performed on original values, percentile or frequency. Decision of categorization technique is based on business understanding. For example, we can categorize income in three categories, namely: High, Average and Low. We can also perform co-variate binning which depends on the value of more than one variables.

What is Feature / Variable Creation & its Benefits?

Feature / Variable creation is a process to generate a new variables / features based on existing variable(s). For example, say, we have date(dd-mm-yy) as an input variable in a data set. We can generate new variables like day, month, year, week, weekday that may have better relationship with target variable. This step is used to highlight the hidden relationship in a variable:

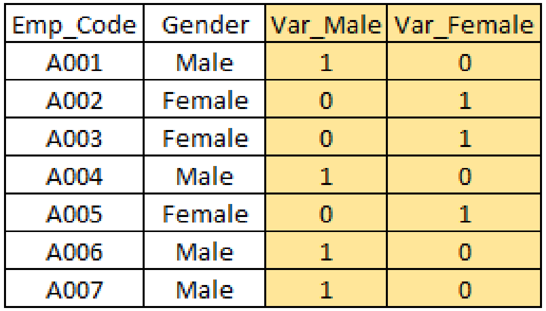


Emp_Code	Gender	Date	New_Day	New_Month	New_Year
A001	Male	21-Sep-11	21	9	2011
A002	Female	27-Feb-13	27	2	2013
A003	Female	14-Nov-12	14	11	2012
A004	Male	07-Apr-13	7	4	2013
A005	Female	21-Jan-11	21	1	2011
A006	Male	26-Apr-13	26	4	2013
A007	Male	15-Mar-12	15	3	2012

There are various techniques to create new features. Let's look at the some of the commonly used methods:

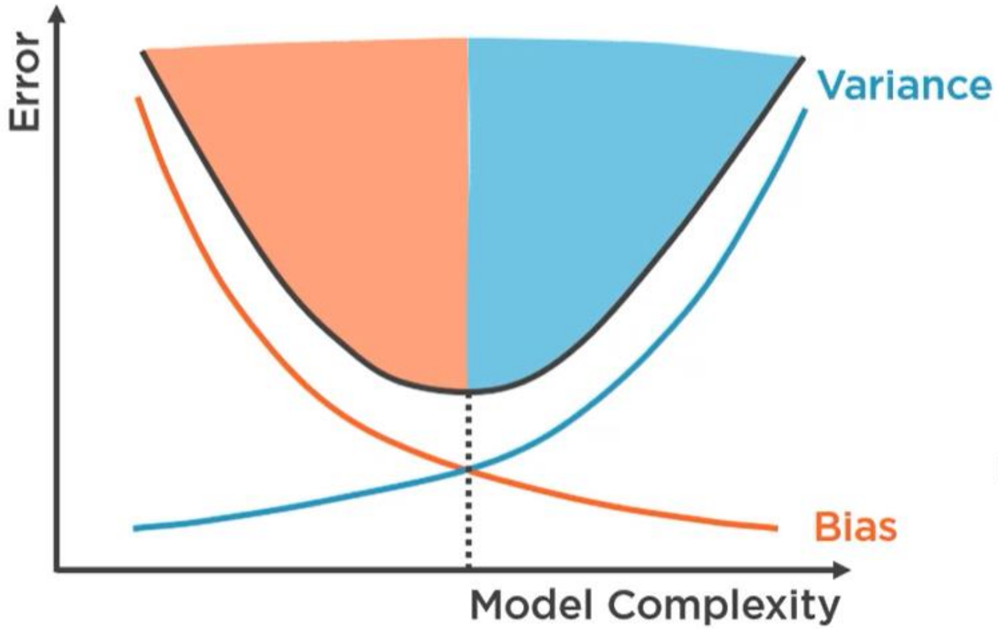
•**Creating derived variables:** This refers to creating new variables from existing variable(s) using set of functions or different methods. In Titanic Dataset , variable age has missing values. To predict missing values, we used the salutation (Master, Mr, Miss, Mrs) of name as a new variable. Ex. Deriving month, day, hour from date column; deriving pass or fail or grades from students marks. How do we decide which variable to create? Honestly, this depends on business understanding of the analyst, his curiosity and the set of hypothesis he might have about the problem.

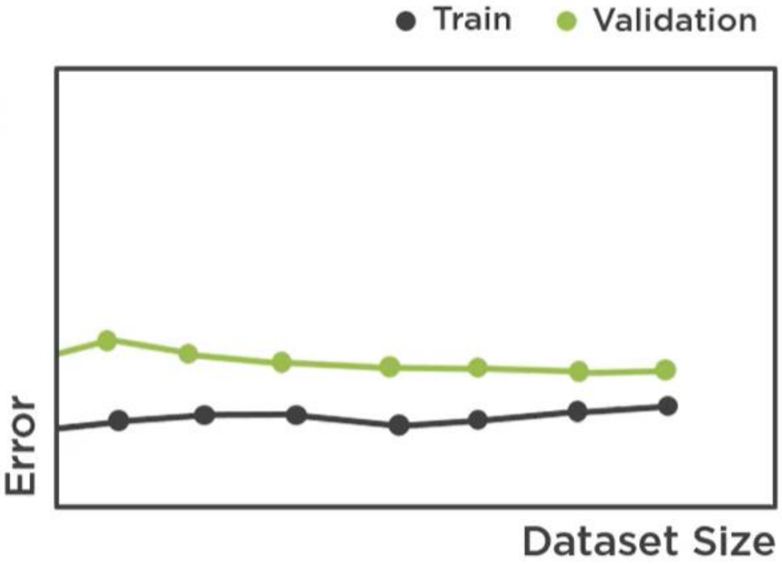
•**Creating dummy variables:** One of the most common application of dummy variable is to convert categorical variable into numerical variables. Dummy variables are also called Indicator Variables. It is useful to take categorical variable as a predictor in statistical models. Categorical variable can take values 0 and 1. Let's take a variable 'gender'. We can produce two variables, namely, "**Var_Male**" with values 1 (Male) and 0 (No male) and "**Var_Female**" with values 1 (Female) and 0 (No Female). We can also create dummy variables for more than two classes of a categorical variables with n or n-1 dummy variables.



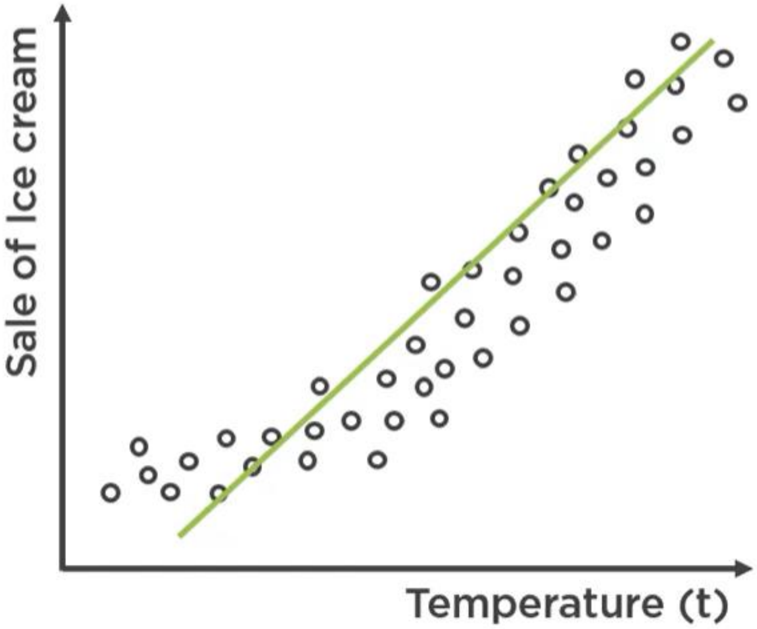
Emp_Code	Gender	Var_Male	Var_Female
A001	Male	1	0
A002	Female	0	1
A003	Female	0	1
A004	Male	1	0
A005	Female	0	1
A006	Male	1	0
A007	Male	1	0

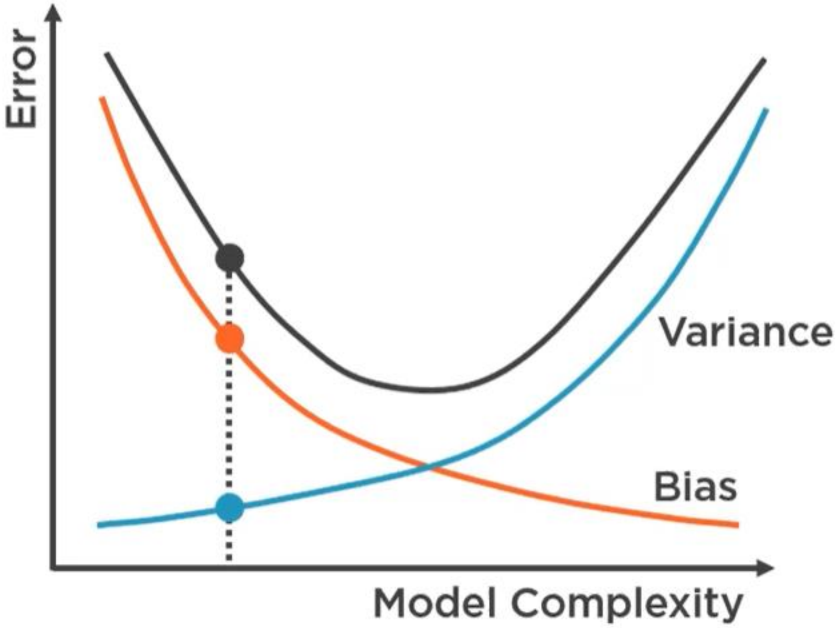
Model Complexity



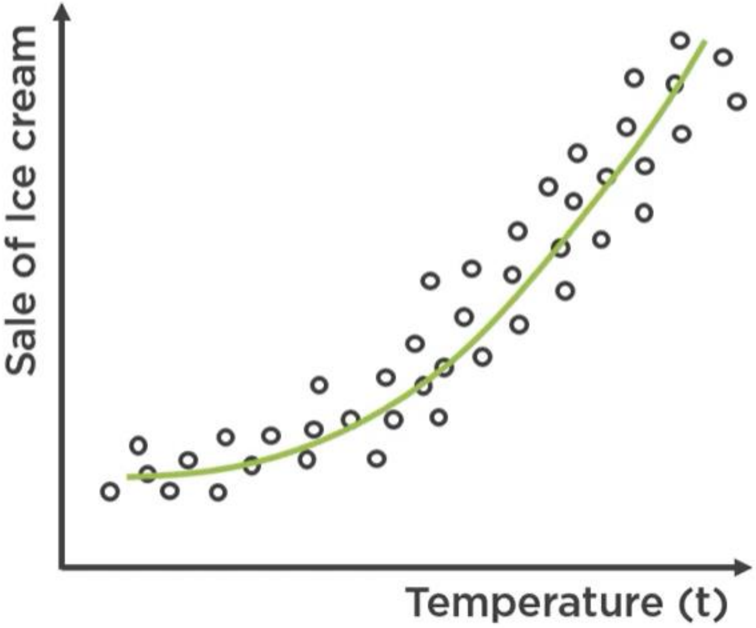


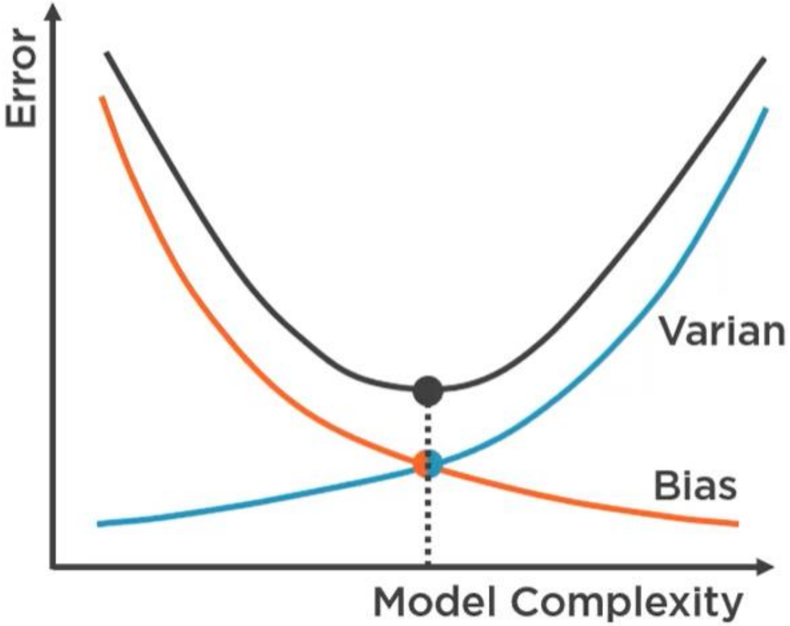
Building Linear Models





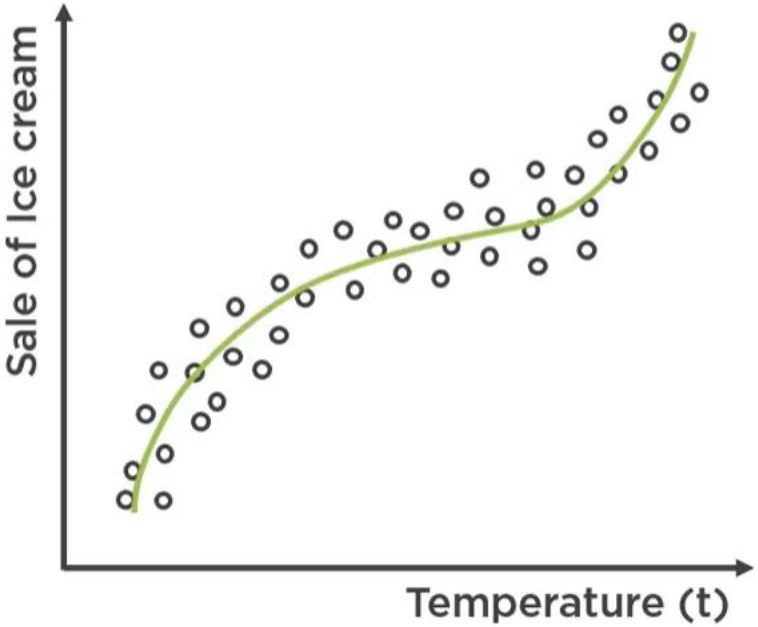
Building Linear Models





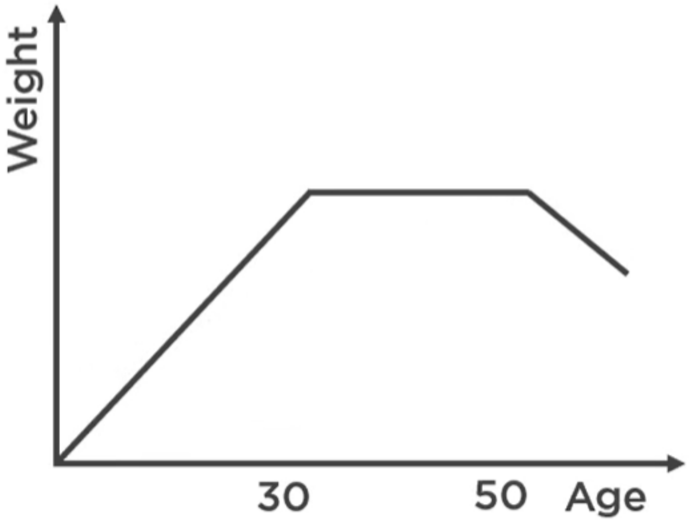
Sale = $a_1(t) + a_2(t)^2 + C$

Building Linear Models



$$\text{Sale} = a_1(t) + a_2(t)^2 + a_3(t)^3 + C$$

Why Binning?



Happy Learning

Thank You !

References: Multiple E-Books/E-Sources